Dataset Description

Dataset Description

The dataset utilized for this project is a comprehensive collection of individual loan application records. The primary objective is to conduct a thorough data preprocessing workflow to prepare the dataset for predictive modeling, specifically for loan approval classification. The dataset provides a multi-faceted view of each applicant, encompassing demographic, financial, and loan-specific attributes.

Key variables include the applicant's personal details, such as age (person_age), gender (person_gender), and education level (person_education). The dataset also provides a robust financial profile for each individual, detailing their annual income (person_income), employment experience in years (person_emp_exp), and home ownership status (person_home_ownership). To assess creditworthiness, the dataset contains information on the length of their credit history (cb_person_cred_hist_length), their assigned credit_score, and any history of previous loan defaults (previous_loan_defaults_on_file). Finally, the loan-specific details are outlined, including the requested loan amount (loan_amnt), the stated purpose of the loan (loan_intent), and the final determination of the application, captured in the loan_status column.

Project Implementation Detail

Task 1: Data Loading and Initial Exploration

Description of the task:

The initial and most fundamental step of this project was to load the loan application dataset into the R environment. The goal of this task was to perform a preliminary investigation to understand the dataset's basic structure and identify any immediate data quality issues. To achieve this, I loaded the data from an Excel file and then used several key functions to inspect its dimensions, column data types, and statistical summary. The final part of this initial exploration was to check for and quantify the number of missing values in each column, which is a critical step that informs the subsequent data-cleaning process.

Code for task 1:

```
install.packages(c("readxl", "dplyr"))

library(readxl)

library(dplyr)
```

data <- read_excel("C:\\Users\\Mehebub Hasan\\Documents\\Data Science project\\Project\\loan_data.xlsx")

str(data)

summary(data)

colSums(is.na(data))

Output from str(data):

```
tibble [201 × 14] (S3: tbl_df/tbl/data.frame)
 $ person_age                    : num [1:201] 21 21 25 23 24 NA 22 24 22 21 ...
 $ person_gender                 : chr [1:201] "female" "female" "female" "female" ...
 $ person_education              : chr [1:201] "Master" "High School" "High School" "Bachelor" ...
 $ person_income                 : num [1:201] 71948 12282 12438 79753 66135 ...
 $ person_emp_exp                : num [1:201] 0 0 3 0 1 0 1 5 3 0 ...
 $ person_home_ownership         : chr [1:201] "RENT" "OWN" "MORTGAGE" "RENT" ...
 $ loan_amnt                     : num [1:201] 35000 1000 5500 35000 35000 2500 35000 35000 35000 1600 ...
 $ loan_intent                   : chr [1:201] "PERSONAL" "EDUCATION" "MEDICAL" "MEDICAL" ...
 $ loan_int_rate                 : num [1:201] 16 11.1 12.9 15.2 14.3 ...
 $ loan_percent_income           : num [1:201] 0.49 NA 0 0.44 0.53 0.19 0.37 0.37 0.35 0.13 ...
 $ cb_person_cred_hist_length    : num [1:201] 3 2 3 2 4 2 3 4 2 3 ...
 $ credit_score                  : num [1:201] 561 504 635 675 586 532 701 585 544 640 ...
 $ previous_loan_defaults_on_file: chr [1:201] "No" "Yes" "No" "No" ...
 $ loan_status                   : num [1:201] 1 0 1 1 1 1 1 1 NA 1 ...
```

Output from summary(data):

```
> summary(data) # Get a statistical summary of each column
   person_age     person_gender      person_education   person_income     person_emp_exp    person_home_ownership
 Min.   : 21.00   Length:201         Length:201         Min.   :  12282   Min.   :  0.000   Length:201
 1st Qu.: 22.00   Class :character   Class :character   1st Qu.:  60501   1st Qu.:  0.000   Class :character
 Median : 23.00   Mode  :character   Mode  :character   Median :  85284   Median :  1.000   Mode  :character
 Mean   : 27.39                                         Mean   : 149875   Mean   :  2.761
 3rd Qu.: 25.00                                         3rd Qu.: 241060   3rd Qu.:  3.000
 Max.   :350.00                                         Max.   :3138998   Max.   :125.000
 NA's   :4                                              NA's   :4
   loan_amnt      loan_intent        loan_int_rate   loan_percent_income cb_person_cred_hist_length  credit_score
 Min.   : 1000   Length:201         Min.   : 5.42   Min.   :0.0000      Min.   :2.00                Min.   :484.0
 1st Qu.:10000   Class :character   1st Qu.:10.65   1st Qu.:0.0900      1st Qu.:2.00                1st Qu.:595.0
 Median :25000   Mode  :character   Median :11.83   Median :0.2350      Median :3.00                Median :630.0
 Mean   :20553                      Mean   :12.29   Mean   :0.2293      Mean   :2.99                Mean   :628.5
 3rd Qu.:28000                      3rd Qu.:14.42   3rd Qu.:0.3425      3rd Qu.:4.00                3rd Qu.:665.0
 Max.   :35000                      Max.   :20.00   Max.   :0.5300      Max.   :4.00                Max.   :807.0
                                                    NA's   :1
 previous_loan_defaults_on_file  loan_status
 Length:201                     Min.   :0.0000
 Class :character               1st Qu.:0.0000
 Mode  :character               Median :1.0000
                                Mean   :0.6162
                                3rd Qu.:1.0000
                                Max.   :1.0000
                                NA's   :3
```

Output from colSums(is.na(data)):

```
> colSums(is.na(data))
                    person_age                  person_gender               person_education                  person_income
                             4                              4                              2                              4
                person_emp_exp          person_home_ownership                      loan_amnt                    loan_intent
                             0                              0                              0                              0
                 loan_int_rate            loan_percent_income     cb_person_cred_hist_length                   credit_score
                             0                              1                              0                              0
previous_loan_defaults_on_file                    loan_status
                             0                              3
```

Code description of task 1:

The execution of this task was accomplished using several core R functions. The read_excel() function from the readxl library was used to import the data into a dataframe named data. Subsequently, the str() function was employed to provide a compact summary of the dataframe's structure, allowing for a quick verification of column names and their respective data types (e.g., numeric, character). To gain initial statistical insights, the summary() function was used to generate descriptive statistics for each variable. Finally, the colSums(is.na()) function was applied to the dataframe to produce a precise count of missing values per column, thereby confirming the necessity for the data imputation steps that follow.

Description of the task 2:

Following the initial data exploration, the next critical task was to address the missing values identified in several key columns. Missing data can lead to biased or inaccurate analysis, so it was essential to apply appropriate imputation techniques. The strategy was to fill in the missing entries using calculated estimates based on the existing data. For numerical columns like person_age and person_income, I used measures of central tendency (mean and median). For categorical columns such as person_gender and person_education, the most frequently occurring value (mode) was used to ensure the integrity of the dataset.

Code for task 2:

```
data_clean <- data


data_clean$person_age[is.na(data_clean$person_age)] <-
round(mean(data_clean$person_age, na.rm = TRUE))

data_clean$person_income[is.na(data_clean$person_income)] <-
median(data_clean$person_income, na.rm = TRUE)

data_clean$loan_percent_income[is.na(data_clean$loan_percent_income)] <-
mean(data_clean$loan_percent_income, na.rm = TRUE)

data_clean$loan_status[is.na(data_clean$loan_status)] <-
as.numeric(names(which.max(table(data_clean$loan_status))))

data_clean$person_gender[is.na(data_clean$person_gender)] <-
names(which.max(table(data_clean$person_gender)))

mode_education <- names(which.max(table(data_clean$person_education)))

data_clean$person_education[is.na(data_clean$person_education)] <- mode_education

colSums(is.na(data_clean))
```

output from colSums(is.na(data_clean)):

```
> colSums(is.na(data_clean))
                person_age              person_gender           person_education              person_income
                         0                          0                          0                          0
            person_emp_exp      person_home_ownership                  loan_amnt                loan_intent
                         0                          0                          0                          0
             loan_int_rate        loan_percent_income cb_person_cred_hist_length               credit_score
                         0                          1                          0                          0
previous_loan_defaults_on_file                loan_status
                         0                          0
```

Code description of task 2:

To perform the data imputation, we first created a copy of the original dataset named data_clean to preserve the raw data. For numerical columns, We applied different strategies based on the data's characteristics; for person_age, We used the round() and mean() functions to fill missing values with the average age, ensuring the column remained in an integer format. For the potentially skewed person_income column, We used the median() function, as it is less sensitive to outliers. For categorical columns like person_gender and person_education, We calculated the mode (the most frequent value) using table() and which.max() and used this to fill the missing entries. Finally, We ran colSums(is.na()) again to confirm that the imputation was successful and that no missing values remained in the dataset.

Task 3: Outlier Detection and Treatment

Description of the task:

After handling missing values, the next step was to identify and treat outliers within the dataset. Outliers are extreme data points that can disproportionately skew statistical measures and negatively impact the performance of predictive models. The primary method for this task involved visualizing the distribution of numerical columns, such as person_age and person_income, using boxplots. These plots provide a clear visual indication of data points that fall far outside the typical range. The Interquartile Range (IQR) method was then employed to programmatically define the boundaries for normal data. Any data points falling outside these boundaries were replaced with the median value of the column, a robust measure that mitigates the influence of these extreme values on the dataset.

Code for task 3:

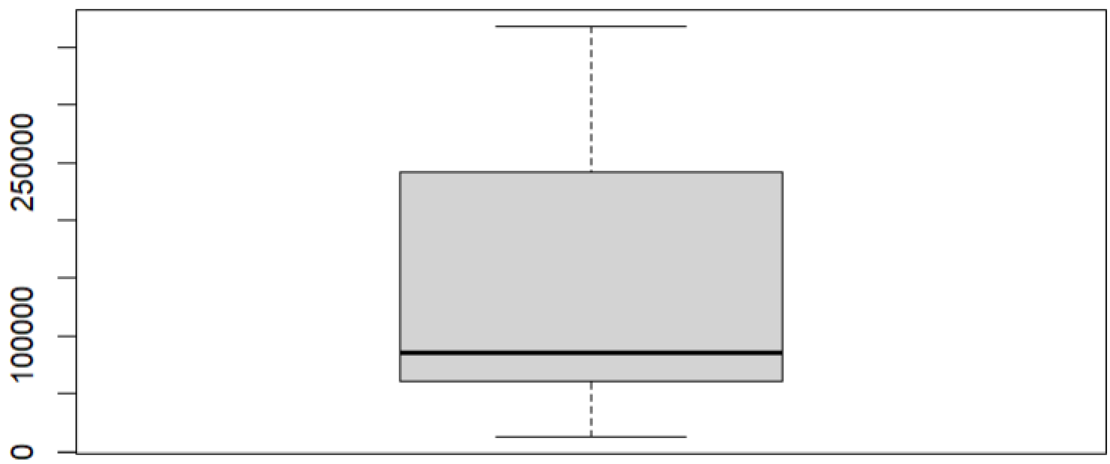
```
Q1_income <- quantile(data_clean$person_income, 0.25, na.rm = TRUE)

Q3_income <- quantile(data_clean$person_income, 0.75, na.rm = TRUE)

IQR_income <- Q3_income - Q1_income


lower_bound_income <- Q1_income - 1.5 * IQR_income

upper_bound_income <- Q3_income + 1.5 * IQR_income
```

median_income <- median(data_clean$person_income, na.rm = TRUE)

data_clean$person_income[data_clean$person_income < lower_bound_income |
data_clean$person_income > upper_bound_income] <- median_income

boxplot(data_clean$person_income)

output for task 3:



Code description of task 3:

The outlier treatment process began with a visual inspection using the boxplot() function to display the distribution of the person_income column. To quantitatively identify outliers, the Interquartile Range was calculated using the quantile() and IQR() functions. The standard statistical rule was applied, defining outliers as any data point below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR. These outliers were then programmatically replaced with the column's median(), chosen for its resilience to extreme values. The process concluded by generating a second boxplot() to visually confirm that the outliers had been successfully treated and the column's distribution was now more condensed. This same procedure was repeated for other key numerical columns, including person_age, person_emp_exp, and credit_score.

Task 4: Data Transformation and Normalization

Description of the task:

This task focused on refining the dataset through several key transformations. The primary objectives were to improve data consistency, convert variables into a machine-learning-friendly format, and scale numerical data to a standard range. The process involved cleaning noisy categorical data by standardizing inconsistent values in the person_home_ownership column. Subsequently, the categorical person_gender variable was converted into a binary numeric format. Finally, to prevent features with wide-ranging values from disproportionately influencing the analysis, Min-Max normalization was applied to the person_income column, rescaling its values to a uniform range between 0 and 1.

Code for task 4:

```
data_clean <- data_clean %>%
  mutate(
    person_home_ownership_clean = case_when(
      tolower(as.character(person_home_ownership)) %in% c("rent", "rentt") ~ "RENT",
      tolower(as.character(person_home_ownership)) %in% c("own", "oown") ~ "OWN",
      TRUE ~ as.character(person_home_ownership)
    )
  )
data_clean$person_home_ownership <- data_clean$person_home_ownership_clean
data_clean$person_home_ownership_clean <- NULL


data_clean$gender_numeric <- ifelse(data_clean$person_gender == "male", 1, 0)


data_clean$income_normalized <- (data_clean$person_income - min(data_clean$person_income))
                    (max(data_clean$person_income) - min(data_clean$person_income))
head(data_clean[c("person_home_ownership", "gender_numeric", "person_income", "income_normalized")])
```

Output for task 4:

```
> head(filtered_data)
# A tibble: 6 × 16
  person_age person_gender person_education person_income person_emp_exp person_home_ownership
       <dbl> <chr>         <chr>                    <dbl>          <dbl> <chr>
1         27 male          Master                  130713              0 RENT
2         26 male          Bachelor                360977              5 RENT
3         26 female        High School             316466              6 RENT
4        350 male          Associate                15229              1 RENT
5         26 male          Associate               133372              1 RENT
6         26 male          Associate               256862              2 RENT
```

Code description of task 4:

The data transformation phase was executed using a combination of base R functions and tools from the dplyr package. To standardize the person_home_ownership column, a mutate() and case_when() workflow was implemented to programmatically correct inconsistent string values. The ifelse() function was then used to efficiently convert the categorical person_gender column into a new binary numeric variable, gender_numeric. Finally, Min-Max normalization was applied to the person_income column by implementing its mathematical formulasubtracting the column's minimum value and dividing by its range to scale all values to a consistent 0-to-1 range. The head() function was used to display a sample of the key transformed columns, providing a clear verification that all operations were successful.

Task 5: Data Preparation for Modeling

Description of the task:

This task involved the final preparatory steps to ensure the dataset was robust, unbiased, and properly structured for a machine learning context. The process included three key operations: first, I identified and removed any complete duplicate rows to prevent data redundancy. Second, I addressed the significant class imbalance in the target variable, loan_status, by applying an undersampling technique to create a balanced dataset. This is a crucial step to prevent a model from being biased towards the majority class. Finally, I partitioned the balanced dataset into a training set (70%) and a testing set (30%), a standard practice that allows a model to be trained on one portion of the data and evaluated on another, unseen portion.

Code for task 5:

```
sum(duplicated(data_clean))

data_clean <- distinct(data_clean)

sum(duplicated(data_clean))

table(data_clean$loan_status)

min_class_size <- min(table(data_clean$loan_status))

balanced_data <- data_clean %>%

  group_by(loan_status) %>%

  slice_sample(n = min_class_size) %>%

  ungroup()
```

```
table(balanced_data$loan_status)

install.packages("caTools")

library(caTools)

set.seed(123)

split <- sample.split(balanced_data$loan_status, SplitRatio = 0.7)

train_data <- subset(balanced_data, split == TRUE)

test_data <- subset(balanced_data, split == FALSE)


cat("Rows in Training Data:", nrow(train_data), "\n")

cat("Rows in Testing Data:", nrow(test_data), "\n")
```

Output from code task 5:

Output from duplicate check:

```
> sum(duplicated(data_clean))
[1] 1
```

```
> sum(duplicated(data_clean))
[1] 0
```

Output from balancing check:

```
> table(data_clean$loan_status)

  0   1
 76 124
```

```
> table(balanced_data$loan_status)

 0  1
76 76
```

Output from data split:

```
> cat("Rows in Training Data:", nrow(train_data), "\n")
Rows in Training Data: 106
> cat("Rows in Testing Data:", nrow(test_data), "\n")
Rows in Testing Data: 46
```

Code description of task 5:

The execution of this task began with the duplicated() function to identify and quantify duplicate entries, which were subsequently removed using the distinct() function from the **dplyr** library. To address class imbalance, I first used table() to inspect the distribution of the loan_status variable. I then implemented an undersampling strategy using **dplyr**'s group_by()

and slice_sample() functions to create a new balanced_data dataframe where both classes were equally represented. For the final step, the caTools library was employed. I set a set.seed(123) to ensure the split is reproducible, and then used the sample.split() function with a SplitRatio of 0.7 to partition the balanced_data into train_data and test_data sets for future model training and evaluation.

Task 6: Descriptive Statistical Analysis

Description of the task:

The final task of the project was to perform a descriptive statistical analysis on the cleaned dataset. The objective was to derive key insights into the data's underlying characteristics by computing measures of both central tendency and spread. For two key numeric variables, person_age and person_income, We calculated the mean, median, and mode to understand the "typical" values. For two categorical variables, person_education and loan_intent, We identified the mode to find the most common categories. Additionally, I calculated measures of spread—including the range, variance, and standard deviation—for the numeric variables to understand their variability and distribution. This analysis is crucial for interpreting the dataset and providing context for any future modeling efforts.

Code for task 6:

Central Tendency code:

```
mean_age <- mean(data_clean$person_age)
cat("Mean Age:", mean_age, "\n")
median_age <- median(data_clean$person_age)
cat("Median Age:", median_age, "\n")


get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}


mode_age <- get_mode(data_clean$person_age)
cat("Mode Age:", mode_age, "\n")
mean_income <- mean(data_clean$person_income)
cat("Mean Income:", mean_income, "\n")
```

```
median_income <- median(data_clean$person_income)
cat("Median Income:", median_income, "\n")

get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

mode_income <- get_mode(data_clean$person_income)
cat("Mode Income:", mode_income, "\n")

mode_education <- get_mode(data_clean$person_education)
cat("Mode Education Level:", mode_education, "\n")
table(data_clean$person_education)
mode_intent <- get_mode(data_clean$loan_intent)
cat("Mode Loan Intent:", mode_intent, "\n")
```

Spread Analysis code:

```
age_range <- max(data_clean$person_age) - min(data_clean$person_age)
cat("Age Range:", age_range, "\n")
age_iqr <- IQR(data_clean$person_age)
cat("Age IQR:", age_iqr, "\n")
age_variance <- var(data_clean$person_age)
cat("Age Variance:", age_variance, "\n")
age_sd <- sd(data_clean$person_age)
cat("Age Standard Deviation:", age_sd, "\n")
income_range <- max(data_clean$person_income) - min(data_clean$person_income)
cat("Income Range:", income_range, "\n")
income_iqr <- IQR(data_clean$person_income)
```

cat("Income IQR:", income_iqr, "\n")

income_variance <- var(data_clean$person_income)

cat("Income Variance:", income_variance, "\n")

income_sd <- sd(data_clean$person_income)

cat("Income Standard Deviation:", income_sd, "\n")

Output for task 6:

Central Tendency Output:

```
> cat("Mean Age:", mean_age, "\n")
Mean Age: 23.53

> cat("Median Age:", median_age, "\n")
Median Age: 23

> cat("Mode Age:", mode_age, "\n")
Mode Age: 22

> cat("Mean Income:", mean_income, "\n")
Mean Income: 133668.5

> cat("Median Income:", median_income, "\n")
Median Income: 85284

> cat("Mode Income:", mode_income, "\n")
Mode Income: 85284

> table(data_clean$person_education)

  Associate    Bachelor   Doctorate High School      Master 
         46          72           1          58          23 

> cat("Mode Loan Intent:", mode_intent, "\n")
Mode Loan Intent: EDUCATION
```

Spread Analysis Output:

```
> cat("Age Range:", age_range, "\n")
Age Range: 6

> cat("Age IQR:", age_iqr, "\n")
Age IQR: 3

> cat("Age Variance:", age_variance, "\n")
Age Variance: 2.742814
> cat("Age Standard Deviation:", age_sd, "\n")
Age Standard Deviation: 1.656144
```

```
> cat("Income Range:", income_range, "\n")
Income Range: 355833
> cat("Income IQR:", income_iqr, "\n")
Income IQR: 180349
> cat("Income Variance:", income_variance, "\n")
Income Variance: 11159706746
> cat("Income Standard Deviation:", income_sd, "\n")
Income Standard Deviation: 105639.5
> cat("Age Range:", age_range, "\n")
Age Range: 6
```

Code description of task 6:

The descriptive analysis was performed using several base R functions. Measures of central tendency were calculated using mean(), median(), and a custom get_mode() function. Measures of spread were calculated using max(), min(), IQR(), var(), and sd().

The interpretation of these results reveals key characteristics of the dataset. For the person_age column, the corrected standard deviation is very low (1.66), indicating that the applicants' ages are highly concentrated around the mean of ~24 years with minimal variation. In sharp contrast, the person_income column shows an extremely high standard deviation (105,639.5) and a mean that is significantly larger than the median, which points to a right-skewed distribution where a few high-income individuals pull the average up. This suggests that for income, the median is a more representative measure of a "typical" applicant. The categorical analysis identified that the most frequent applicant has a Bachelor's degree and is applying for a loan for "Education".

Project Code

```
install.packages(c("readxl", "dplyr"))

install.packages("caTools")

library(caTools)

library(readxl)

library(dplyr)

install.packages("dplyr")

library(dplyr)


data <- read_excel("C:\\Users\\Mehebub Hasan\\Documents\\Data Science project\\Project\\loan_data.xlsx")
```

```
str(data)
summary(data)
colSums(is.na(data))

data_clean <- data

colSums(is.na(data_clean))

data_clean$person_age[is.na(data_clean$person_age)] <-
  round(mean(data_clean$person_age, na.rm = TRUE))

data_clean$person_income[is.na(data_clean$person_income)] <-
  median(data_clean$person_income, na.rm = TRUE)

data_clean$loan_percent_income[is.na(data_clean$loan_percent_income)] <-
  mean(data_clean$loan_percent_income, na.rm = TRUE)

data_clean$loan_status[is.na(data_clean$loan_status)] <-
  as.numeric(names(which.max(table(data_clean$loan_status))))

data_clean$person_gender[is.na(data_clean$person_gender)] <-
  names(which.max(table(data_clean$person_gender)))

mode_education <- names(which.max(table(data_clean$person_education)))
data_clean$person_education[is.na(data_clean$person_education)] <- mode_education

colSums(is.na(data_clean))


boxplot(data_clean$person_age)
```

```
quantile(data_clean$person_age)


Q1_age <- quantile(data_clean$person_age, 0.25)
Q3_age <- quantile(data_clean$person_age, 0.75)
IQR_age <- Q3_age - Q1_age


lower_bound_age <- Q1_age - 1.5 * IQR_age
upper_bound_age <- Q3_age + 1.5 * IQR_age



median_age <- median(data_clean$person_age)
data_clean$person_age[data_clean$person_age < lower_bound_age | data_clean$person_age
> upper_bound_age] <- median_age


boxplot(data_clean$person_age)


Q1_income <- quantile(data_clean$person_income, 0.25, na.rm = TRUE)
Q3_income <- quantile(data_clean$person_income, 0.75, na.rm = TRUE)
IQR_income <- Q3_income - Q1_income


lower_bound_income <- Q1_income - 1.5 * IQR_income
upper_bound_income <- Q3_income + 1.5 * IQR_income


median_income <- median(data_clean$person_income, na.rm = TRUE)
data_clean$person_income[data_clean$person_income < lower_bound_income |
data_clean$person_income > upper_bound_income] <- median_income


boxplot(data_clean$person_income)


boxplot(data_clean$person_emp_exp)
Q1_exp <- quantile(data_clean$person_emp_exp, 0.25, na.rm = TRUE)
```

```
Q3_exp <- quantile(data_clean$person_emp_exp, 0.75, na.rm = TRUE)
IQR_exp <- Q3_exp - Q1_exp

lower_bound_exp <- Q1_exp - 1.5 * IQR_exp
upper_bound_exp <- Q3_exp + 1.5 * IQR_exp

median_exp <- median(data_clean$person_emp_exp, na.rm = TRUE)
data_clean$person_emp_exp[data_clean$person_emp_exp < lower_bound_exp |
data_clean$person_emp_exp > upper_bound_exp] <- median_exp

boxplot(data_clean$person_emp_exp)
median_exp <- median(data_clean$person_emp_exp, na.rm = TRUE)
data_clean$person_emp_exp[data_clean$person_emp_exp < lower_bound_exp |
data_clean$person_emp_exp > upper_bound_exp] <- median_exp


boxplot(data_clean$person_emp_exp)

boxplot(data_clean$credit_score)
Q1_score <- quantile(data_clean$credit_score, 0.25, na.rm = TRUE)
Q3_score <- quantile(data_clean$credit_score, 0.75, na.rm = TRUE)
IQR_score <- Q3_score - Q1_score

lower_bound_score <- Q1_score - 1.5 * IQR_score
upper_bound_score <- Q3_score + 1.5 * IQR_score


median_score <- median(data_clean$credit_score, na.rm = TRUE)
data_clean$credit_score[data_clean$credit_score < lower_bound_score |
data_clean$credit_score > upper_bound_score] <- median_score
boxplot(data_clean$credit_score)
```

```
data_clean <- data_clean %>%
  mutate(
    person_home_ownership_clean = case_when(
      tolower(as.character(person_home_ownership)) %in% c("rent", "rentt") ~ "RENT",
      tolower(as.character(person_home_ownership)) %in% c("own", "oown") ~ "OWN", # Changed "OOWN" to "oown"
      TRUE ~ as.character(person_home_ownership)
    )
  )

data_clean$person_home_ownership <- data_clean$person_home_ownership_clean
data_clean$person_home_ownership_clean <- NULL
table(data_clean$person_home_ownership)
data_clean$person_home_ownership_clean <- NULL

unique(data_clean$person_home_ownership)

data_clean$gender_numeric <- ifelse(data_clean$person_gender == "male", 1, 0)

head(data_clean[c("person_gender", "gender_numeric")])

data_clean$income_normalized <- (data_clean$person_income - min(data_clean$person_income)) /
  (max(data_clean$person_income) - min(data_clean$person_income))

head(data_clean[c("person_income", "income_normalized")])

summary(data_clean$income_normalized)
```

```
sum(duplicated(data_clean))

data_clean <- distinct(data_clean)

sum(duplicated(data_clean))

filtered_data <- data_clean %>%
  filter(person_age > 25, loan_intent == "EDUCATION")

nrow(filtered_data)

head(filtered_data)

table(data_clean$loan_status)

min_class_size <- min(table(data_clean$loan_status))

balanced_data <- data_clean %>%
  group_by(loan_status) %>%
  slice_sample(n = min_class_size) %>%
  ungroup()

table(balanced_data$loan_status)

set.seed(123)

split <- sample.split(balanced_data$loan_status, SplitRatio = 0.7)

train_data <- subset(balanced_data, split == TRUE)
```

```
test_data <- subset(balanced_data, split == FALSE)

cat("Rows in Training Data:", nrow(train_data), "\n")
cat("Rows in Testing Data:", nrow(test_data), "\n")

mean_age <- mean(data_clean$person_age)
cat("Mean Age:", mean_age, "\n")

median_age <- median(data_clean$person_age)
cat("Median Age:", median_age, "\n")

get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

mode_age <- get_mode(data_clean$person_age)
cat("Mode Age:", mode_age, "\n")



mean_income <- mean(data_clean$person_income)
cat("Mean Income:", mean_income, "\n")

median_income <- median(data_clean$person_income)
cat("Median Income:", median_income, "\n")

get_mode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
# Calculate the mode
mode_income <- get_mode(data_clean$person_income)
cat("Mode Income:", mode_income, "\n")

mode_education <- get_mode(data_clean$person_education)
cat("Mode Education Level:", mode_education, "\n")

table(data_clean$person_education)

mode_intent <- get_mode(data_clean$loan_intent)
cat("Mode Loan Intent:", mode_intent, "\n")

table(data_clean$loan_intent)

age_range <- max(data_clean$person_age) - min(data_clean$person_age)
cat("Age Range:", age_range, "\n")

age_iqr <- IQR(data_clean$person_age)
cat("Age IQR:", age_iqr, "\n")

age_variance <- var(data_clean$person_age)
cat("Age Variance:", age_variance, "\n")

age_sd <- sd(data_clean$person_age)
cat("Age Standard Deviation:", age_sd, "\n")

income_range <- max(data_clean$person_income) - min(data_clean$person_income)
cat("Income Range:", income_range, "\n")
```

```
income_iqr <- IQR(data_clean$person_income)
cat("Income IQR:", income_iqr, "\n")

income_variance <- var(data_clean$person_income)
cat("Income Variance:", income_variance, "\n")

income_sd <- sd(data_clean$person_income)
cat("Income Standard Deviation:", income_sd, "\n")
```