**AMERICAN INTERNATIONAL**A close up of a sign

Description automatically generated

**UNIVERSITY-BANGLADESH**

**Faculty of Science and Technology**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignment Title: | Midterm Project | | | |
| Assignment No: | 01 | | Date of Submission: | 14 December 2024 |
| Course Title: | Introduction to Data Science | | | |
| Course Code: | CSC4180 | | Section: | C |
| Semester: | Fall | 2024-25 | Course Teacher: | TOHEDUL ISLAM |

**Declaration and Statement of Authorship:**

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student’s work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the faculty for review and comparison, including review by external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

*\* Student(s) must complete all details except the faculty use part.*

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

|  |  |
| --- | --- |
| Group Name/No.: | 14 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **ID** | **Program** | **Signature** |
| 01 | AZMINUR RAHMAN | 22-46459-1 | BSc [CSE] |  |
| 02 | MD. ABDUL MALEK RONY | 20-43687-2 | BSc [CSE] |  |
| 03 | MD. TAMIM | 22-46918-1 | BSc [CSE] |  |
| 04 | MD. FAHIM RAHMAN | 21-45303-2 | BSc [CSE] |  |

|  |  |  |
| --- | --- | --- |
| ***Faculty use only*** | | |
| FACULTY COMMENTS | **Marks Obtained** |  |
|  |
|  |
|  | **Total Marks** |  |
|  |
|  |

Dataset: **Introduction**

This dataset contains financial and personal information of 201 individuals, useful for analyzing loan applications and predicting loan repayment behavior. The dataset includes 14 attributes:

* **person\_age**: Age of the individual.
* **person\_gender**: Gender of the individual.
* **person\_education**: Education level of the individual.
* **person\_income**: Annual income of the individual.
* **person\_emp\_exp**: Employment experience in years.
* **person\_home\_ownership**: Home ownership status (e.g., RENT, OWN, MORTGAGE).
* **loan\_amnt**: Loan amount requested.
* **loan\_intent**: Purpose of the loan (e.g., PERSONAL, EDUCATION, MEDICAL).
* **loan\_int\_rate**: Interest rate on the loan.
* **loan\_percent\_income**: Percentage of income allocated to loan repayment.
* **cb\_person\_cred\_hist\_length**: Length of the individual's credit history in years.
* **credit\_score**: Credit score of the individual.
* **previous\_loan\_defaults\_on\_file**: Indicates if there are previous loan defaults (Yes/No).
* **loan\_status**: Outcome of the loan application (e.g., 1 for approved, 0 for denied).

While comprehensive, the dataset has some missing values in attributes like **person\_age**, **person\_income**, **person\_education**, and **loan\_status**. There are also potential inconsistencies, such as a typo in the **person\_home\_ownership** column (e.g., "RENTT"). These issues make the dataset an excellent candidate for preprocessing, data cleaning, and exploratory data analysis tasks.

Dataset: **About data**

* **Library Use:**

library(readxl)

library(dplyr)

* **Read Data**

mydata <- read\_excel("C:/Users/AZMINUR RAHMAN/OneDrive - American International University-Bangladesh/2024-2025, Fall/INTRODUCTION TO DATA SCIENCE [C]/Mid/Lab/Project/Midterm\_Dataset\_Section(C).xlsx", sheet = "Sheet1")

View(mydata)

str(mydata)

summary(mydata)

num\_instances <- nrow(mydata)

num\_attributes <- ncol(mydata)

print(paste("Number of instances (rows):", num\_instances))

print(paste("Number of columns:", num\_attributes))

missing\_values\_indices <- lapply(mydata, function(x) {

if (is.numeric(x) || is.character(x)) {

return(which(is.na(x) | x == ""))

} else {

return(NULL)

}

})

print(missing\_values\_indices)

|  |
| --- |
|  |
|  |
|  |
|  |

**Description:** Load Dataset to mydata. Found the total number of row and column using ncol() and nrow() function.Then we found all the missing values with the help of is.numeric() and is.character() function. Used print() functions to show the output in one line.

Dataset: **Data Preparation & Exploration**

**Column: person\_age**

* **Missing Value Imputation: Replacing NA Values with Median**

age\_median <- round(median(mydata$person\_age, na.rm = TRUE))

mydata$person\_age[is.na(mydata$person\_age)] <- age\_median

* **Outlier Detection and Removal with Interquartile Range (IQR) Method**

Q1 <- quantile(mydata$person\_age, 0.25, na.rm = TRUE)

Q3 <- quantile(mydata$person\_age, 0.75, na.rm = TRUE)

IQR\_value <- Q3 - Q1

threshold <- 1.5

outlier\_condition <- (mydata$person\_age < (Q1 - threshold \* IQR\_value)) |

(mydata$person\_age > (Q3 + threshold \* IQR\_value))

* **Serial Update After Removing Outliers**

mydata <- mydata %>%

filter(!outlier\_condition) %>%

arrange(row\_number())

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** First, we replaced the NA values with the median. Then with the help of IQR method we have found the Outliers then removed those Outliers. To update the rows after removing outliers we have used pipe operator (%>%)

**Column: person\_gender**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_values\_gender <- unique(mydata$person\_gender)

print(unique\_values\_gender)

**Output:**

|  |
| --- |
|  |

* **Data conversion: Converting categorical attributes to numeric (Gender is a categorical data)**

mydata$person\_gender <- tolower(mydata$person\_gender)

mydata$person\_gender <- factor(mydata$person\_gender,

levels = c("male", "female"),

labels = c(1, 2))

View(mydata)

**Output:**

|  |
| --- |
|  |

* **Missing Value Imputation: Replacing NA Values with Mode**

mode\_gender <- as.numeric(names(sort(table(mydata$person\_gender), decreasing = TRUE)[1]))

mydata$person\_gender[is.na(mydata$person\_gender)] <- mode\_gender

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: person\_education**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_education <- unique(mydata$person\_education)

print(unique\_education)

**Output:**

|  |
| --- |
|  |

* **Data conversion: Converting categorical attributes to numeric**

mydata$person\_education <- tolower(mydata$person\_education)

mydata$person\_education <- factor(mydata$person\_education,

levels = c("master", "high school", "bachelor", "associate", "doctorate"),

labels = c(1, 2, 3, 4, 5))

View(mydata)

**Output:**

|  |
| --- |
|  |

* **Missing Value Imputation: Replacing NA Values with Mode**

mode\_education <- names(which.max(table(mydata$person\_education)))

mydata$person\_education[is.na(mydata$person\_education)] <- mode\_education

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: person\_income**

* **Missing Value Imputation: Replacing NA Values with Median**

income\_median <- median(mydata$person\_income, na.rm = TRUE)

mydata$person\_income[is.na(mydata$person\_income)] <- income\_median

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** Using median() value for replacing NA.

**Column: person\_emp\_exp**

* **The ‘person\_emp\_exp’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_emp\_exp <- sum(is.na(mydata$person\_emp\_exp))

print(missing\_values\_emp\_exp)

**Output:**

|  |
| --- |
|  |

* **Outlier Detection and Removal with Interquartile Range (IQR) Method**

Q1 <- quantile(mydata$person\_emp\_exp, 0.25, na.rm = TRUE)

Q3 <- quantile(mydata$person\_emp\_exp, 0.75, na.rm = TRUE)

IQR\_value <- Q3 - Q1

threshold <- 1.5

outlier\_condition <- (mydata$person\_emp\_exp < (Q1 - threshold \* IQR\_value))|

(mydata$person\_emp\_exp > (Q3 + threshold \* IQR\_value))

* **Serial Update After Removing Outliers**

mydata <- mydata %>%

filter(!outlier\_condition) %>%

arrange(row\_number())

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** By the help of IQR method we have found the Outliers then removed those Outliers. To update the rows after removing outliers we have used pipe operator (%>%)

**Column: person\_home\_ownership**

* **Detecting and Recovering Noisy Values:**

unique\_home\_ownership <- unique(mydata$person\_home\_ownership)

print(unique\_home\_ownership)

**Output:**

|  |
| --- |
|  |

* **Data conversion: Converting categorical attributes to numeric and** **replaces all instances of "rentt" with "rent" and “oown” with “own”**

mydata$person\_home\_ownership <- tolower(mydata$person\_home\_ownership)

mydata$person\_home\_ownership <- ifelse(mydata$person\_home\_ownership == "rentt", "rent", mydata$person\_home\_ownership)

mydata$person\_home\_ownership <- ifelse(mydata$person\_home\_ownership == "oown", "own", mydata$person\_home\_ownership)

mydata$person\_home\_ownership <- factor(mydata$person\_home\_ownership,

levels = c("rent", "own", "mortgage", "other"),

labels = c(1, 2, 3, 4))

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the unique values using unique() then we replaces all instances of "rentt" with "rent" and “oown” with “own” by using ifelse(). At last we have converted categorical attributes to numeric with the help of factor function.

**Column: loan\_amnt**

* **The ‘loan\_amnt’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_loan\_amnt <- sum(is.na(mydata$loan\_amnt))

print(missing\_values\_loan\_amnt)

**Output:**

|  |
| --- |
|  |

**Column: loan\_intent**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_loan\_intent <- unique(mydata$loan\_intent)

print(unique\_loan\_intent)

**Output:**

|  |
| --- |
|  |

* **Data conversion: Converting categorical attributes to numeric**

mydata$loan\_intent <- tolower(mydata$loan\_intent)

mydata$loan\_intent <- factor(mydata$loan\_intent,

levels = c("personal", "education", "medical", "venture", "homeimprovement", "debtconsolidation"),

labels = c(1, 2, 3, 4, 5, 6))

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: loan\_int\_rate**

* **The ‘loan\_int\_rate’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_loan\_amnt <- sum(is.na(mydata$loan\_amnt))

print(missing\_values\_loan\_amnt)

**Output:**

|  |
| --- |
|  |

**Column: loan\_percent\_income**

* **Missing Value Imputation: Replacing NA Values with Median**

loan\_percent\_income\_median <- median(mydata$loan\_percent\_income, na.rm = TRUE)

mydata$loan\_percent\_income[is.na(mydata$loan\_percent\_income)] <- loan\_percent\_income\_median

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** Using median() value for replacing NA.

**Column: cb\_person\_cred\_hist\_length**

* **The ‘cb\_person\_cred\_hist\_length’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_cb\_person\_cred\_hist\_length <- sum(is.na(mydata$cb\_person\_cred\_hist\_length))

print(missing\_values\_cb\_person\_cred\_hist\_length)

**Output:**

|  |
| --- |
|  |

**Column: credit\_score**

* **The ‘credit\_score’ column exhibits optimal data quality with no missing or invalid values.**

missing\_values\_credit\_score <- sum(is.na(mydata$credit\_score))

print(missing\_values\_credit\_score)

**Output:**

|  |
| --- |
|  |

**Column: previous\_loan\_defaults\_on\_file**

* **Detecting and Recovering Noisy Values: There is no Noisy values**

unique\_previous\_loan\_defaults\_on\_file <- unique(mydata$previous\_loan\_defaults\_on\_file)

print(unique\_previous\_loan\_defaults\_on\_file)

**Output:**

|  |
| --- |
|  |

* **Data conversion: Converting categorical attributes to numeric**

mydata$previous\_loan\_defaults\_on\_file <- tolower(mydata$previous\_loan\_defaults\_on\_file)

mydata$previous\_loan\_defaults\_on\_file <- factor(mydata$previous\_loan\_defaults\_on\_file,

levels = c("yes", "no"),

labels = c(1, 2))

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the unique values using unique() then we have converted categorical attributes to numeric with the help of factor function.

**Column: loan\_status**

* **Checking missing value.**

missing\_values\_loan\_status <- sum(is.na(mydata$loan\_status))

print(missing\_values\_loan\_status)

**Output:**

|  |
| --- |
|  |

* **Missing Value Imputation: Replacing NA Values with Mode**

mode\_loan\_status <- as.numeric(names(sort(table(mydata$loan\_status), decreasing = TRUE)[1]))

mydata$loan\_status[is.na(mydata$loan\_status)] <- mode\_loan\_status

View(mydata)

**Output:**

|  |
| --- |
|  |

**Description:** At first we found the missing values quantity by using sum() & is.na() then we replace the missing values by mode.

**Remove duplicate rows:**

**Code:**

duplicate\_rows <- mydata[duplicated(mydata), ]

print(paste("Number of duplicate rows:", nrow(duplicate\_rows)))

mydata <- mydata[!duplicated(mydata), ]

View(mydata)

**Output:**

|  |
| --- |
|  |
|  |

**Description:** Finding duplicated rows by using duplicated() function and drop those rows.

**Apply min max for all numeric column:**

**Code:**

numeric\_columns <- sapply(mydata, is.numeric)

mydata[numeric\_columns] <- lapply(mydata[numeric\_columns], function(x) {

(x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) - min(x, na.rm = TRUE))

})

**Output:**

|  |
| --- |
|  |

**Description:** Finding duplicated rows by using duplicated() function and drop those rows.

**Date Visualization**

* **Find Missing Values:**

**Code:**

na\_counts <- colSums(is.na(mydata))

print(na\_counts)

**Visualization:**

barplot(na\_counts, names.arg = names(na\_counts),

ylab = "Number of Missing Values", col = "red", cex.names = 0.9,

main = "Missing Values per Attribute", las = 2)

**Output:**

|  |
| --- |
|  |
|  |
|  |

**Description:** Using colSums() and is.na() we found the number of missing values for each attribute.

* **Find Outliers:**

|  |
| --- |
|  |
|  |
|  |

**Description:** Using boxplot, we found outliers on Age and Person\_emp\_exp we no need to find other attributes outliers.

**After removing outliers:**

|  |
| --- |
|  |
|  |
|  |

* **Summary:**

|  |
| --- |
|  |

* **Mean-Median-Mode:**

**Code:**

descriptive\_stats <- function(column) {

mean\_value <- mean(column, na.rm = TRUE)

median\_value <- median(column, na.rm = TRUE)

mode\_value <- as.numeric(names(sort(table(column), decreasing = TRUE)[1]))

return(c(Mean = mean\_value, Median = median\_value, Mode = mode\_value))

}

descriptive\_summary <- lapply(mydata[numeric\_columns], descriptive\_stats)

descriptive\_summary <- do.call(rbind, descriptive\_summary)

print("Descriptive Statistics for Numeric Columns:")

print(descriptive\_summary)

**Vizualization:**

|  |
| --- |
| A screenshot of a computer screen  Description automatically generated |

**Description:** A function, descriptive\_stats, is defined to calculate the mean, median, and mode of a given numeric column, with missing values handled using na.rm = TRUE. The mean(), median(), and table() functions are used to compute these statistics, and the results are returned as a named vector. The function is applied to all numeric columns of a dataset (mydata) using lapply(), and the outputs are combined into a single data frame using do.call(rbind, ...). Finally, the descriptive statistics for the numeric columns are displayed.

* **Mean-Median-Mode Graph:**

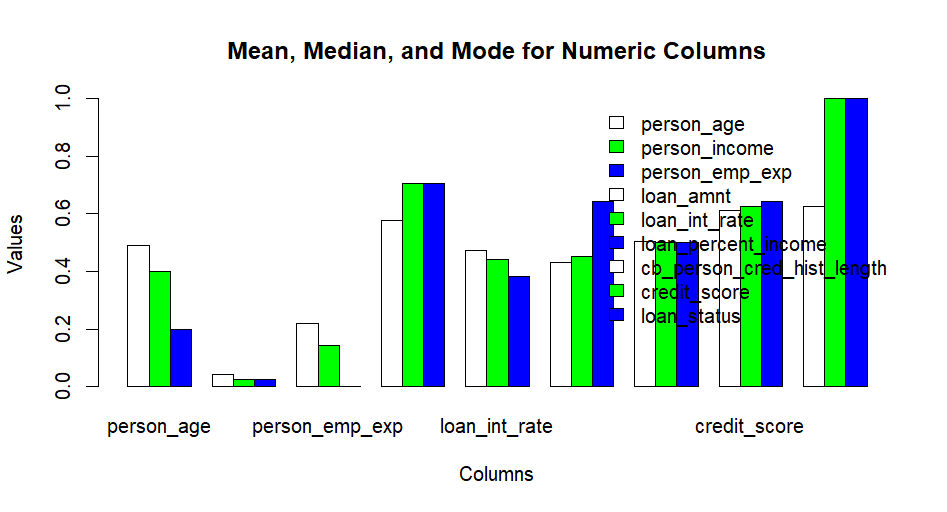
**Code:**

barplot(t(descriptive\_summary), beside = TRUE, col = c("white", "green", "blue"),

legend.text = rownames(descriptive\_summary), args.legend = list(x = "topright", bty = "n"),

main = "Mean, Median, and Mode for Numeric Columns", ylab = "Values", xlab = "Columns")

**Vizualization:**



**Description:** Barplot has been drawn by using barplot() function to visualize.

**Final Data Set:**

A screen shot of a computer

Description automatically generated

This is the outcome of the data set after cleaning all the data.

THE END