IMAAD IMRAN HAJWANE

202101132 / 23

DMPM LAB

ASSIGNMENT 01

STATEMENT**:**

Data Exploration & Visualization

1. Read the dataset file that is supplied to you.

2. Identify the variables in the file and determine whether any variable has any missing values.

3. Input some of the variables that have missing values using their corresponding mean values. Verify whether your task has been correctly done.

4. Determine the "summary" information for the numerical variables.

5. Identify the "distributions" of the numerical variables and plot the distributions.

6. Transform the numeric variables into their natural log values and scale [0 - 1] values.

7. Check whether the numeric variables follow normality conditions.

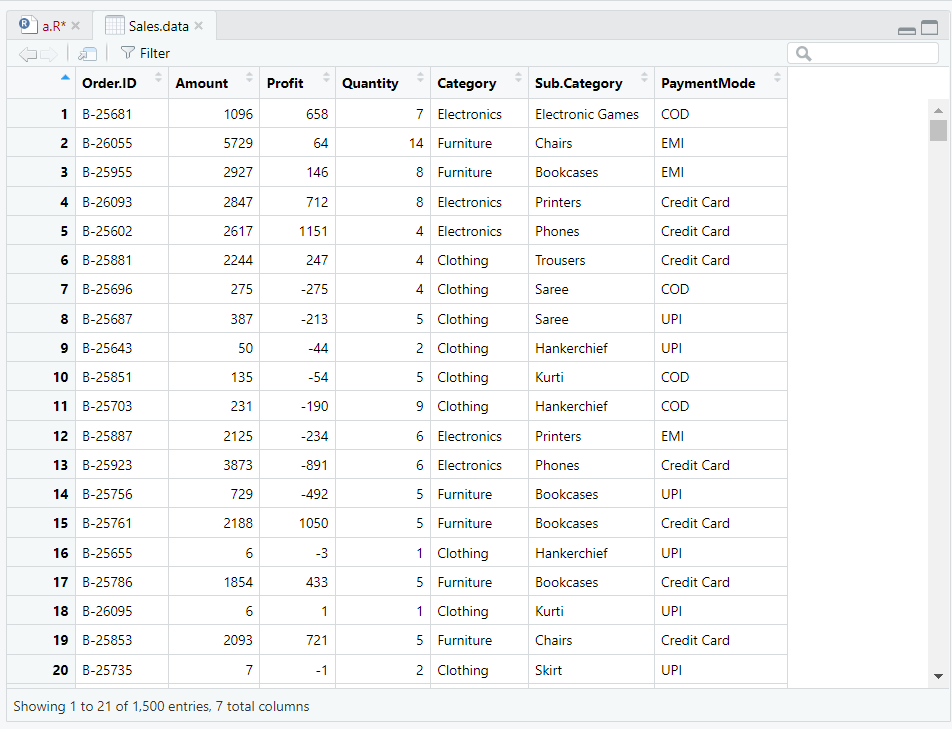
8. Find the correlation matrix for all the variables in the dataset and plot the graph of the correlation matrix.

9. Any additional ways of Data Exploration & Visualization will be highly appreciated.

SOLUTIONS:

Sales.data <- read.csv ("C:/Users/iamim/OneDrive/Desktop/Sixth Semester/DMPM\_LAB/A1/Sales data.csv")

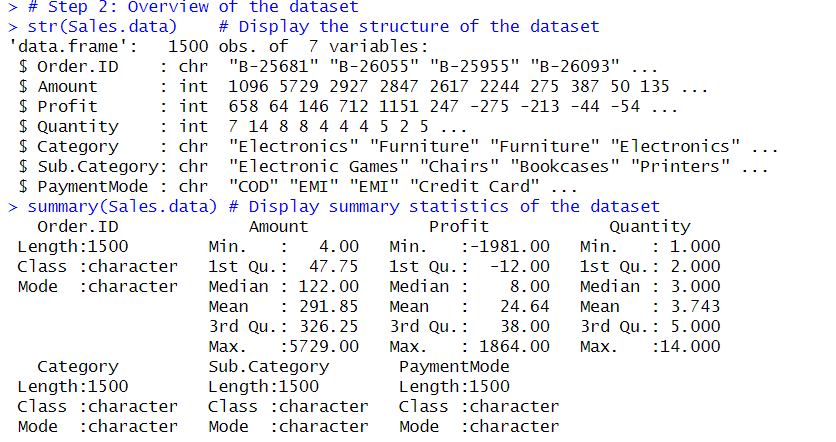
View (Sales.data)



2.

str (Sales.data) # Display the structure of the dataset

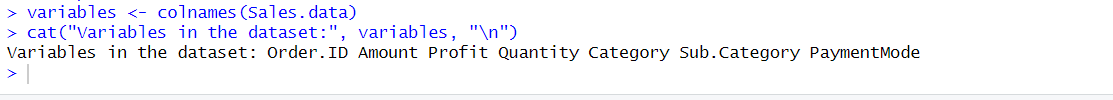
summary (Sales.data) # Display summary statistics of the dataset



#Get variable names

variables <- colnames (Sales.data)

cat ("Variables in the dataset:", variables, "\n")

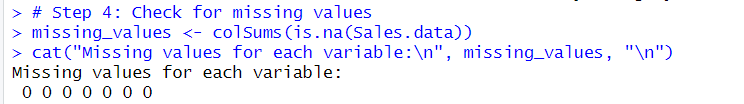




# Check for missing values

missing\_values <- colSums(is.na (Sales.data))

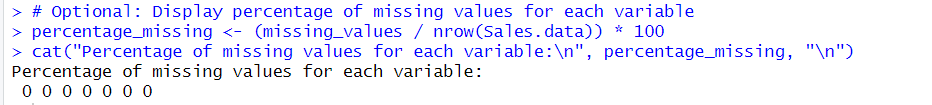
cat ("Missing values for each variable:\n", missing\_values, "\n")



# Display percentage of missing values for each variable

percentage\_missing <- (missing\_values / nrow(Sales.data)) \* 100

cat ("Percentage of missing values for each variable:\n", percentage\_missing, "\n")









# Select only numerical variables

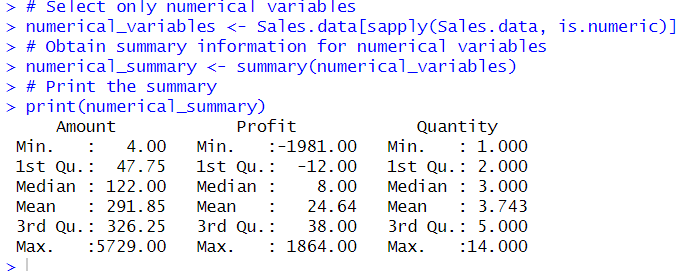
> numerical\_variables <- Sales.data[sapply(Sales.data, is.numeric)]

> # Obtain summary information for numerical variables

> numerical\_summary <- summary(numerical\_variables)

> # Print the summary

> print(numerical\_summary)





# Select only numerical variables

numerical\_variables <- Sales.data[sapply(Sales.data, is.numeric)]

# Plot histograms for numerical variables

par(mfrow = c(2, 2)) # Setting up a 2x2 grid for subplots

for (variable in colnames(numerical\_variables)) {

hist(numerical\_variables[[variable]], main = paste("Histogram of", variable), col = "lightblue", border = "black", xlab = variable)

}

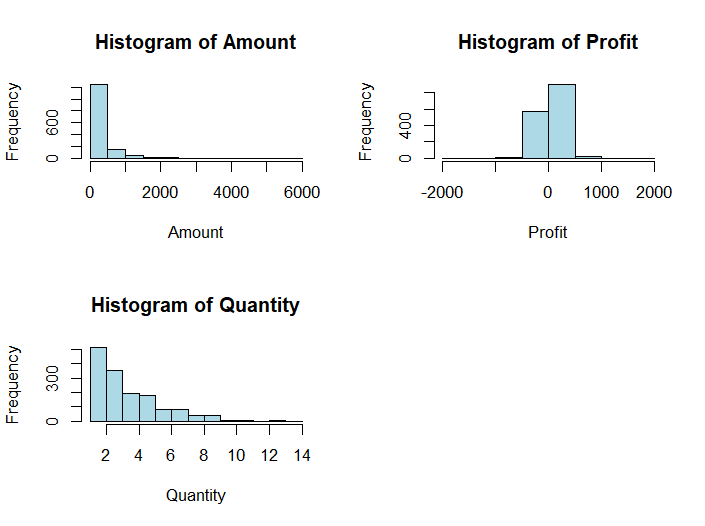
# Plot density plots for numerical variables

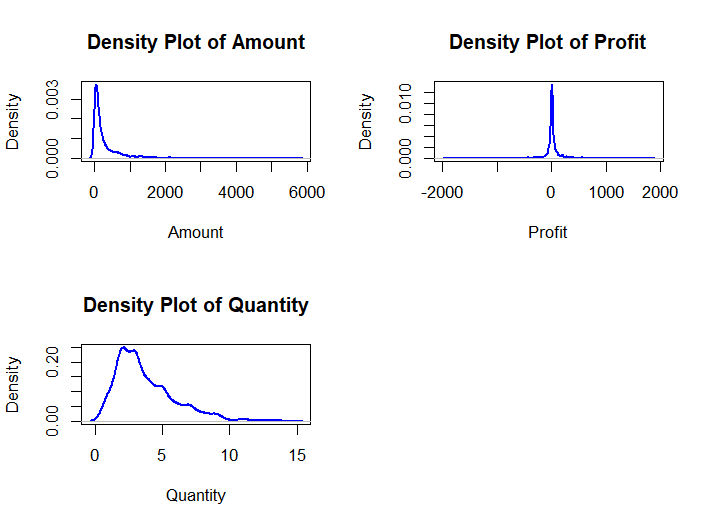
par(mfrow = c(2, 2)) # Resetting the layout

for (variable in colnames(numerical\_variables)) {

plot(density(numerical\_variables[[variable]]), main = paste("Density Plot of", variable), col = "blue", lwd = 2, xlab = variable)

}







# Assuming 'Sales.data' is your dataset

# Replace 'YourNumericVariable1', 'YourNumericVariable2', etc. with the actual numerical variable names in your dataset

# Select only numerical variables

numerical\_variables <- Sales.data[sapply(Sales.data, is.numeric)]

# Log transformation

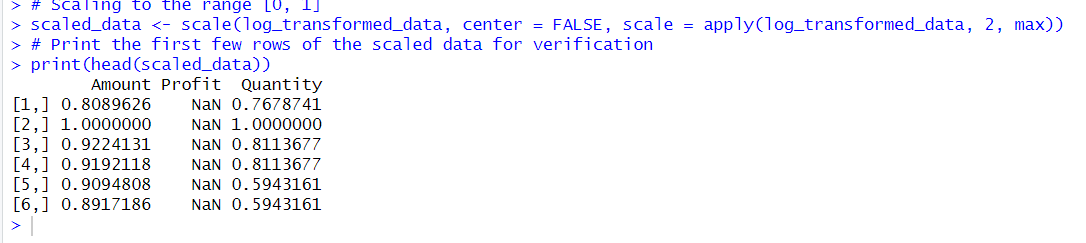
log\_transformed\_data <- log(numerical\_variables + 1) # Adding 1 to avoid log(0) issues

# Scaling to the range [0, 1]

scaled\_data <- scale(log\_transformed\_data, center = FALSE, scale = apply(log\_transformed\_data, 2, max))

# Print the first few rows of the scaled data for verification

print(head(scaled\_data))





# Select only transformed and scaled numeric variables

scaled\_numerical\_variables <- scaled\_data

# Shapiro-Wilk Test

shapiro\_test\_results <- sapply(scaled\_numerical\_variables, function(x) shapiro.test(x)$p.value)

# Q-Q Plots

par(mfrow = c(2, 2)) # Setting up a 2x2 grid for subplots

for (variable in colnames(scaled\_numerical\_variables)) {

qqnorm(scaled\_numerical\_variables[[variable]], main = paste("Q-Q Plot of", variable))

qqline(scaled\_numerical\_variables[[variable]], col = 2)

}

# Print the results of the Shapiro-Wilk Test

print(shapiro\_test\_results)



# Select only numeric variables

numeric\_variables <- Sales.data[sapply(Sales.data, is.numeric)]

# Calculate the correlation matrix

correlation\_matrix <- cor(numeric\_variables, use = "complete.obs")

# Print the correlation matrix

print(correlation\_matrix)

# Plot the graph of the correlation matrix using corrplot

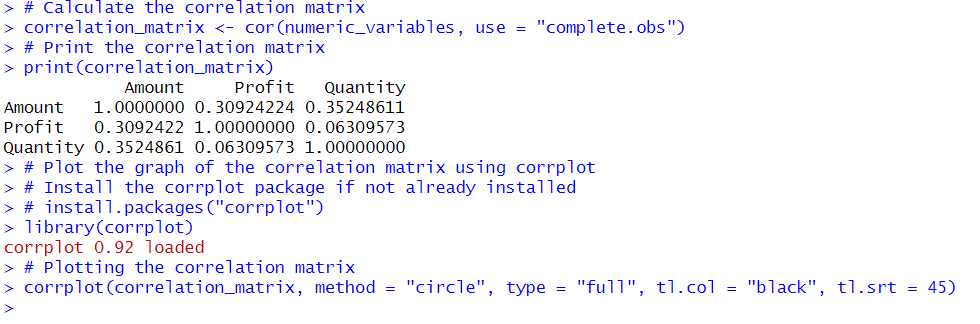
# Install the corrplot package if not already installed

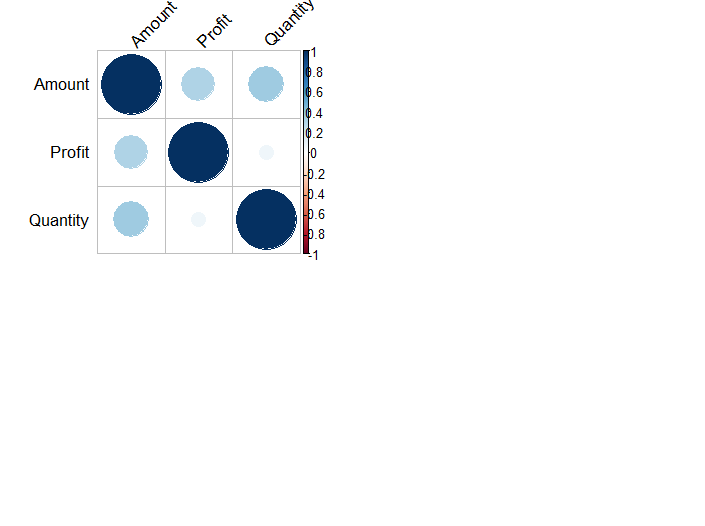
# install.packages("corrplot")

library(corrplot)

# Plotting the correlation matrix

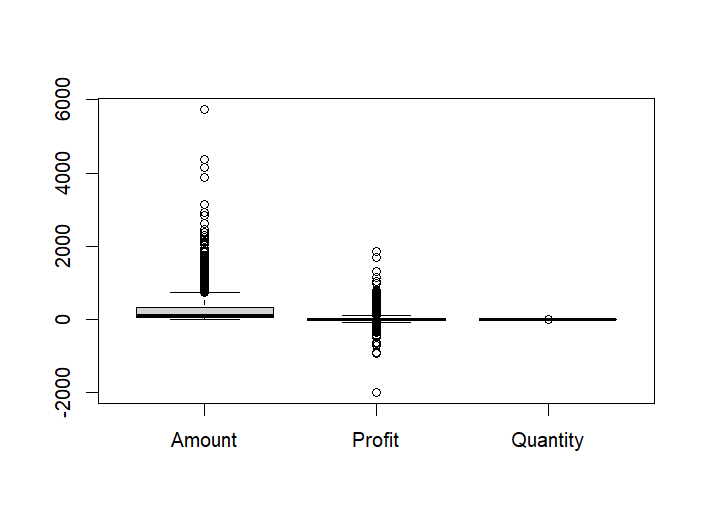
corrplot(correlation\_matrix, method = "circle", type = "full", tl.col = "black", tl.srt = 45)







boxplot(numeric\_variables)



# Assuming 'Category' is a categorical variable

barplot(table(Sales.data$Category), col = "lightblue", main = "Bar Plot of Category")

