Exploratory Data Analysis (EDA) in R is the process of examining datasets to summarize their main characteristics, often with visual methods. The goal of EDA is to understand the structure of the data, identify patterns, detect outliers, and test assumptions before applying any modelling techniques.

Here are the basic steps and techniques for performing EDA in R:

**1. Loading the Data**

Copy code

# Load a dataset (for example, a CSV file)

data <- read.csv("your\_dataset.csv")

# Display the first few rows of the dataset

head(data)

# Display the structure of the dataset

str(data)

# Get summary statistics of the dataset

summary(data)

**2. Checking the Data Structure**

* **str()**: Displays the structure of the dataset, showing each column's data type and some values.
* **summary()**: Provides summary statistics for each variable (mean, median, quartiles, etc.).

Copy code

# Check the structure

str(data)

# Summary of the data

summary(data)

**3. Checking for Missing Values**

You can check if your data contains missing values, which might need to be handled (e.g., with imputation or removal).

r

Copy code

# Check for missing values in the dataset

sum(is.na(data))

# Check missing values by column

colSums(is.na(data))

**4. Univariate Analysis (Single Variable)**

**For Categorical Variables:**

r

Copy code

# Frequency table for a categorical variable

table(data$categorical\_column)

# Bar plot for a categorical variable

barplot(table(data$categorical\_column))

**For Numerical Variables:**

r

Copy code

# Summary statistics of a numerical column

summary(data$numerical\_column)

# Histogram to visualize the distribution

hist(data$numerical\_column, main = "Histogram", xlab = "Values", col = "blue")

# Boxplot to check for outliers

boxplot(data$numerical\_column, main = "Boxplot", horizontal = TRUE)

**5. Bivariate Analysis (Two Variables)**

**For Categorical vs Categorical:**

r

Copy code

# Cross-tabulation (contingency table)

table(data$categorical\_column1, data$categorical\_column2)

# Stacked bar plot for categorical variables

barplot(table(data$categorical\_column1, data$categorical\_column2), beside = TRUE, col = c("red", "blue"))

**For Numerical vs Numerical:**

r

Copy code

# Scatter plot to visualize the relationship between two numeric variables

plot(data$numerical\_column1, data$numerical\_column2, main = "Scatter Plot", xlab = "Variable 1", ylab = "Variable 2")

**For Numerical vs Categorical:**

r

Copy code

# Boxplot to compare a numerical variable across categories

boxplot(data$numerical\_column ~ data$categorical\_column, main = "Boxplot by Category", xlab = "Category", ylab = "Value")

**6. Correlation Analysis (for numerical variables)**

To check the relationship between multiple numerical variables:

r

Copy code

# Compute the correlation matrix

cor(data[, sapply(data, is.numeric)])

# Visualize the correlation matrix

library(corrplot)

corrplot(cor(data[, sapply(data, is.numeric)]), method = "circle")

**7. Detecting Outliers**

You can detect outliers using boxplots and basic statistics:

r

Copy code

# Boxplot for detecting outliers in a numerical column

boxplot(data$numerical\_column, main = "Boxplot for Outlier Detection")

# Using IQR method to detect outliers

Q1 <- quantile(data$numerical\_column, 0.25)

Q3 <- quantile(data$numerical\_column, 0.75)

IQR <- Q3 - Q1

lower\_bound <- Q1 - 1.5 \* IQR

upper\_bound <- Q3 + 1.5 \* IQR

# Values below lower\_bound or above upper\_bound are considered outliers

outliers <- data[data$numerical\_column < lower\_bound | data$numerical\_column > upper\_bound, ]

print(outliers)

**8. Visualizations (using ggplot2)**

For advanced visualizations, the ggplot2 package is widely used.

r

Copy code

# Install ggplot2 if you haven't already

install.packages("ggplot2")

library(ggplot2)

# Histogram using ggplot

ggplot(data, aes(x = numerical\_column)) +

geom\_histogram(binwidth = 5, fill = "blue", color = "black") +

ggtitle("Histogram of Numerical Column")

# Boxplot using ggplot

ggplot(data, aes(x = categorical\_column, y = numerical\_column)) +

geom\_boxplot(fill = "lightblue") +

ggtitle("Boxplot of Numerical Column by Category")

# Scatter plot using ggplot

ggplot(data, aes(x = numerical\_column1, y = numerical\_column2)) +

geom\_point() +

ggtitle("Scatter Plot")

**9. Checking Data Distribution**

You can check the normality of a numeric variable using Q-Q plots:

r

Copy code

# Q-Q plot to check for normality

qqnorm(data$numerical\_column)

qqline(data$numerical\_column, col = "red")

**10. Handling Missing Data**

You can deal with missing data in several ways (e.g., removing or imputing).

r

Copy code

# Remove rows with missing data

data\_cleaned <- na.omit(data)

# Replace missing values with the mean

data$numerical\_column[is.na(data$numerical\_column)] <- mean(data$numerical\_column, na.rm = TRUE)

**Key Functions and Libraries:**

* **Basic functions**: head(), summary(), str(), table(), plot(), hist(), boxplot()
* **ggplot2**: For advanced visualizations.
* **corrplot**: For visualizing correlation matrices.
* **Handling missing values**: na.omit(), is.na(), colSums(is.na())

**Summary:**

* **Univariate analysis** focuses on single variables.
* **Bivariate analysis** focuses on relationships between two variables.
* **Multivariate analysis** can explore relationships between more than two variables.
* **Visualizations** such as histograms, boxplots, and scatter plots are key tools for EDA.