Exploratory Data Analysis (EDA) Using R

Introduction

Exploratory Data Analysis (EDA) is a critical step in the data analysis pipeline. It helps uncover the underlying structure of the data, identify anomalies, and understand the relationships between variables. Exploratory data analysis (EDA) is used by data scientists to analyse and investigate data sets and summarize their main characteristics, often employing data visualization methods. In this tutorial, we will go through a step-by-step guide to performing EDA using R.

Step 1: Load libraries and dataset

For this tutorial, we will use the "mtcars" dataset, which comes preloaded with R. We will also use the ggplot2, dplyr, and corrplot libraries. Install and load them as follows:

```r
# Install required libraries
install.packages("ggplot2")
install.packages("dplyr")
install.packages("corrplot")

# Load libraries
library(ggplot2)
library(dplyr)
library(corrplot)

# Load dataset
data(mtcars)
```

Step 2: Examine the data structure and summary statistics

Before diving into EDA, it's essential to understand the dataset's structure and basic summary statistics.

```r
# View the dataset
head(mtcars)

# Structure of the dataset
str(mtcars)

# Summary statistics
summary(mtcars)
```

Step 3: Data cleaning and preprocessing

Check for missing values and handle them if necessary. For the "mtcars" dataset, there are no missing values.

```r
# Check for missing values
sum(is.na(mtcars))
```

Step 4: Univariate analysis

Univariate analysis explores individual variables through descriptive statistics and visualizations. For continuous variables, use histograms and box plots. For categorical variables, use bar plots.

```r
# Histogram for 'mpg' (miles per gallon)
ggplot(mtcars, aes(x = mpg)) +
  geom_histogram(binwidth = 2, fill = "blue", alpha = 0.7) +
  labs(title = "Histogram of MPG", x = "Miles Per Gallon", y = "Frequency")

# Box plot for 'hp' (horsepower)
ggplot(mtcars, aes(y = hp)) +
  geom_boxplot(fill = "orange", alpha = 0.7) +
  labs(title = "Box Plot of Horsepower", y = "Horsepower")
```

Step 5: Bivariate analysis

Bivariate analysis explores the relationship between two variables. For continuous variables, use scatter plots and correlation matrices. For categorical variables, use stacked bar plots or mosaic plots.

```r
# Scatter plot between 'mpg' and 'hp'
ggplot(mtcars, aes(x = mpg, y = hp)) +
  geom_point(color = "red", alpha = 0.7) +
  labs(title = "Scatter Plot: MPG vs. Horsepower",
       x = "Miles Per Gallon", y = "Horsepower")

# Correlation matrix
cor_matrix <- cor(mtcars)
corrplot(cor_matrix, method = "circle")
```

Step 6: Multivariate analysis

Multivariate analysis explores relationships between multiple variables simultaneously. Use scatter plot matrices, parallel coordinate plots, and heatmap visualizations

```r
# Heatmap for correlation matrix
library(heatmaply)
heatmaply(cor_matrix, labRow = colnames(mtcars),
          labCol = colnames(mtcars),
          title = "Heatmap of Correlation Matrix")
```

Step 7: Outlier detection

Outliers can significantly affect your analysis and models. Identify them using box plots, scatter plots, and statistical tests, such as the IQR method or the Z-score method.

```
# Outlier detection using IQR method for 'hp'
hp_iqr <- IQR(mtcars$hp)
hp_q1 <- quantile(mtcars$hp, 0.25)
hp_q3 <- quantile(mtcars$hp, 0.75)
hp_lower_bound <- hp_q1 - 1.5 * hp_iqr
hp_upper_bound <- hp_q3 + 1.5 * hp_iqr
outliers_hp <- mtcars[mtcars$hp < hp_lower_bound | mtcars$hp > hp_upper_bound,]

# Print outliers in 'hp'
print(outliers_hp)
```

Step 8: Feature transformation

Transforming features can help improve the performance of your models. Log transformation, scaling, and normalization are common techniques.

```
# Log transformation of 'hp'
mtcars$log_hp <- log(mtcars$hp)

# Min-max scaling of 'wt' (weight)
min_wt <- min(mtcars$wt)
max_wt <- max(mtcars$wt)
mtcars$scaled_wt <- (mtcars$wt - min_wt) / (max_wt - min_wt)
```

Conclusion:

In this tutorial, we performed Exploratory Data Analysis (EDA) using R. We covered data loading, cleaning, preprocessing, univariate analysis, bivariate analysis, multivariate analysis, outlier detection, and feature transformation. By following these steps, you can gain insights into your dataset and make informed decisions for further analysis and modelling.

Activity 1: Analysing a different dataset

Choose a new dataset to perform EDA using the steps outlined in this tutorial. You can use any dataset of your choice, or select one from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.php).

1. Load your dataset into R.
2. Perform data cleaning and preprocessing.
3. Conduct univariate, bivariate, and multivariate analyses.
4. Detect and handle outliers.
5. Perform feature transformation if needed.
6. Document your findings and observations.

Activity 2: Categorical variable exploration

The "mtcars" dataset mostly contains continuous variables. Find a dataset with categorical variables (e.g., the Titanic dataset) and explore the relationships between these variables using appropriate visualization techniques, such as bar plots, stacked bar plots, and mosaic plots.

Activity 3: Customizing visualizations

Enhance the visualizations created in this tutorial by customizing them according to your preferences. You can use different colors, shapes, and themes for your plots. Explore the ggplot2 documentation (https://ggplot2.tidyverse.org/reference/) to learn more about available customization options.

1. Modify the histogram and box plot in the univariate analysis section with new colours, fill, and themes.
2. Customize the scatter plot in the bivariate analysis section with different point shapes, sizes, and colours.
3. Change the appearance of the correlation matrix plot and heatmap in the multivariate analysis section.
4. Experiment with different visualization techniques for each analysis step to find the most suitable one for your dataset.