Data Analysis Lab 3 Report, Ilya Ivanov, JAVA

Introduction

In this laboratory assignment, the datasets Iris and Mtcars are used.

The Iris dataset contains 50 samples from each of three Iris species: Iris setosa, Iris virginica, and Iris versicolor. It offers valuable insights into the relationships between the species and the features of the flowers, such as sepal and petal dimensions.
The Mtcars dataset includes various automobile attributes, such as miles per gallon (mpg), the number of cylinders (cyl), horsepower (hp), and others. It offers valuable insights into the relationships between vehicle features like fuel efficiency, engine specifications, weight, and performance metrics. This dataset enables analysis of how different factors impact a car's efficiency, speed, and transmission type, providing a comprehensive view of automotive design and functionality.

The main goal is to perform exploratory data analysis (EDA) and visualization on them by using R:

1. Analyze these datasets;
2. Visualize key properties;
3. Draw conclusions about the relationships between variables.

The following libraries were applied: ggplot2 (Data visualization (plots, graphs)), dplyr (Data manipulation (`filter`, `group_by`)) and tidyverse (Tidy workflows)

Setup

1. Folder Structure: There is a zip format file containing the R file `LAB3_Ilya_Ivanov.R`. The zip must be unzipped;
2. Package Installation: Install the necessary packages such as ggplot2, dplyr and tidyverse for performing EDA and visualization (`library()` function).

Part 1: Load the Datasets

First of all, the datasets are loaded by using `data()` function. Finally, the first 6 rows of each dataset are displayed by applying `head()` function.

Part 2: Exploratory Data Analysis (EDA)

Missing values are checked by implementing the following function:

1. The `%>%` operator, known as the **pipe**, is used to pass the output of one function directly into the next function as its input;
2. Function `summarise_all()` is a `dplyr` function that applies a summary function to **all** columns in the dataset;
3. Lambda Function `~sum(is.na(.))` Returns a **single-row dataframe** where each column contains the count of missing values for that respective column;
4. `unlist()`: Converts the single-row data frame into a **named numeric vector**;
5. `sum()`: Provides the **total number of missing values** in the entire dataset;
6. `cat()`: Displays the total number of missing values.
7. Then, counting and removing duplicate rows in datasets using the tidyverse and leveraging the power of dplyr functions like duplicated() and distinct()are performed.

Result: There are no missing in the datasets. However, one duplicate was found in the `Iris` dataset, which was removed after all.

2.1 Iris Dataset

Likewise, structure of the `Iris` dataset is displayed by `str()` function and summary is performed by applying `summary()` function.

Feature Relations in Iris dataset

Moreover, a **pair plot** to visualize relationships between the species and other features is created. For that the `pairs()` function in R is implemented which creates a **matrix of scatterplots** (also known as a **pair plot**) for each pair of variables in a dataset (this allows you to visualize potential relationships, correlations, and patterns between multiple variables at once).

Result:

1. **'Petal.Length' and 'Petal.Width'** exhibit a strong positive linear relationship, indicating they are highly correlated.
2. **'Sepal.Length' and 'Petal.Length'** show a noticeable positive correlation, though weaker than the petal dimensions.

3. **'Sepal.Width'** has little to no clear correlation with 'Sepal.Length' and other features, appearing almost random.
4. Overall, petal measurements are more strongly correlated than sepal measurements, making them more effective for differentiating `Iris` species.

'Sepal Length' by Species

Also, a **boxplot** to visualize the distribution of **'Sepal.Length'** by species is displayed.

There are several steps to build such a box plot:

1. Initialize the Plot with `ggplot()`;
2. Add the Boxplot Layer with `geom_boxplot()`;
3. Add a Title with `ggtitle()`;
4. Label the Axes with `xlab()` and `ylab()`.

Result:

1. **'Sepal.Length'** increases from Setosa (median ~5 cm) to Versicolor (median ~6 cm) and Virginica (median >6.5 cm).
2. Setosa exhibits low variability with a narrow range, while Virginica shows higher variability and includes an outlier.
3. The clear separation in median 'Sepal.Length' among species makes it an effective feature for distinguishing Iris species.
4. There is one outlier in Virginica, which has length less than 5 cm.

Correlation Matrix of Numerical Variables

Correlation matrix is built to show the strength of relationships between numerical variables. First of all, calculation of the correlation matrix for the numeric variables is performed. Then, a heatmap is built by using the `heatmap()` function. The color scale is set using the `col` parameter.

Result: 'Petal.Width' and '**Petal.Length'/'Sepal.Width'** and **'Sepal.Length'** are one the most strongly correlated variables in the dataset.

2.2 Mtcars Dataset

Summary and structure of `Mtcars` dataset are applied the same way as in `Iris` example

Miles per 'Gallon' and 'Horsepower'

A **scatter plot** showing the relationship between 'miles per gallon (mpg)' and 'horsepower (hp)'.

`ggplot()` is applied again, but combined with `geom_point(color = "blue")` to add a scatter plot layer with points represented in blue.
Result: There is a clear **negative** correlation between 'horsepower' and 'miles per gallon (mpg)' in the `Mtcars` dataset, with higher horsepower linked to lower fuel efficiency. Likewise, cars with lower horsepower (50-100 hp) show a wide mpg range (20-35), while those above 150 hp consistently have mpg below 15. Also, a few high-horsepower vehicles (over 300 hp) are outliers with mpg around as a result, this indicates that increased engine power significantly compromises fuel efficiency in these cars.

Average 'MPG' by Number of Cylinders

Moreover, a **bar graph** is built by using the following functions:

1. `%>%` – Initiate a pipeline to manipulate the `Mtcars` dataset;
2. `group_by(cyl)` – group the dataset by the number of cylinders (`cyl`);
3. `summarise(average_mpg = mean(mpg))` – Calculate the average 'MPG' for each group of cylinders;
4. `ggplot()` combined with `geom_bar(stat = "identity", fill = "orange")` — Create a bar plot where the height of each bar corresponds to the computed average 'MPG' (`stat = "identity"`). The bars are filled with orange color.

Result:

1. **4 Cylinders** have the highest average 'MPG' (over 25), indicating better fuel efficiency.
2. **6 Cylinders** show a moderate average 'MPG' around 20, balancing power and efficiency.
3. **8 Cylinders** have the lowest average 'MPG' below 15, reflecting poorer fuel efficiency due to higher engine power.
4. Finally, as the number of cylinders increases, average fuel efficiency decreases, demonstrating a trade-off between engine power and fuel economy.

Part 3: Statistical Tests

3.1 Iris Dataset: ANOVA Results

For `Iris` dataset `ANOVA` results are calculated by using `aov(Sepal.Length ~ Species, data = iris)` to perform a one-way ANOVA test to assess whether the mean 'Sepal.Length' differs across the three 'Species' groups in the `Iris` dataset.

Result: The one-way `ANOVA` shows that species significantly affects **'Sepal.Length'** (F = 119.3, p < 2e-16). This indicates that Sepal Length differs notably between at least two Iris species

3.2 Mtcars Dataset: Correlation between 'Horsepower' and 'Miles per Gallon'

Correlation between 'HP' and 'MPG' is calculated to find out how strong their relationship is. `cor.test(x, y)` is applied to perform a test.

Result: The correlation between horsepower and 'MPG' in the `Mtcars` dataset is **-0.776** (`p = 1.788e-07`), which shows a **strong**, statistically **significant negative relationship**. Likewise, the 95% confidence interval ranges from **-0.885 to -0.586**, confirming that as horsepower increases, fuel efficiency (mpg) decreases significantly.