**What is Exploratory Data Analysis?**

Last Updated : 13 Jan, 2025

Exploratory Data Analysis (EDA) is an important first step in data science projects. It involves looking at and visualizing data to understand its main features, find patterns, and discover how different parts of the data are connected.

EDA helps to spot any unusual data or outliers and is usually done before starting more detailed statistical analysis or building models. In this article, we will discuss what is **Exploratory Data Analysis (EDA)** and the [steps to perform EDA](https://www.geeksforgeeks.org/steps-for-mastering-exploratory-data-analysis-eda-steps/).

**Why Exploratory Data Analysis is Important?**

Exploratory Data Analysis (EDA) is important for several reasons, especially in the context of data science and statistical modeling. Here are some of the key reasons why EDA is a critical step in the data analysis process:

* Helps to understand the dataset, showing how many features there are, the type of data in each feature, and how the data is spread out, which helps in choosing the right methods for analysis.
* EDA helps to identify hidden patterns and relationships between different data points, which help us in and model building.
* Allows to spot errors or unusual data points (outliers) that could affect your results.
* Insights that you obtain from EDA help you decide which features are most important for building models and how to prepare them to improve performance.
* By understanding the data, EDA helps us in choosing the best modeling techniques and adjusting them for better results.

**Types of Exploratory Data Analysis**

There are various sorts of EDA strategies based on nature of the records. Depending on the number of columns we are analyzing we can divide EDA into three types: [Univariate, bivariate and multivariate](https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis/).

**1. Univariate Analysis**

Univariate analysis focuses on studying one variable to understand its characteristics. It helps describe the data and find patterns within a single feature. Common methods include **histograms** to show data distribution, **box plots** to detect outliers and understand data spread, and **bar charts** for categorical data. **Summary statistics** like[**mean**,](https://www.geeksforgeeks.org/mean/) [**median**](https://www.geeksforgeeks.org/median/), **mode**,[**variance**,](https://www.geeksforgeeks.org/variance/) and [**standard deviation**](https://www.geeksforgeeks.org/standard-deviation-formula/) help describe the central tendency and spread of the data

**2. Bivariate Analysis**

Bivariate analysis focuses on exploring the relationship between two variables to find connections, correlations, and dependencies. It’s an important part of exploratory data analysis that helps understand how two variables interact. Some key techniques used in bivariate analysis include **scatter plots**, which visualize the relationship between two continuous variables; **correlation coefficient**, which measures how strongly two variables are related, commonly using [**Pearson’s correlation**](https://www.geeksforgeeks.org/pearson-correlation-coefficient/) for linear relationships; and **cross-tabulation**, or **contingency tables**, which show the frequency distribution of two categorical variables and help understand their relationship.

[**Line graphs**](https://www.geeksforgeeks.org/line-graph/) are useful for comparing two variables over time, especially in time series data, to identify trends or patterns. [**Covariance**](https://www.geeksforgeeks.org/mathematics-covariance-and-correlation/) measures how two variables change together, though it’s often supplemented by the correlation coefficient for a clearer, more standardized view of the relationship.

**3. Multivariate Analysis**

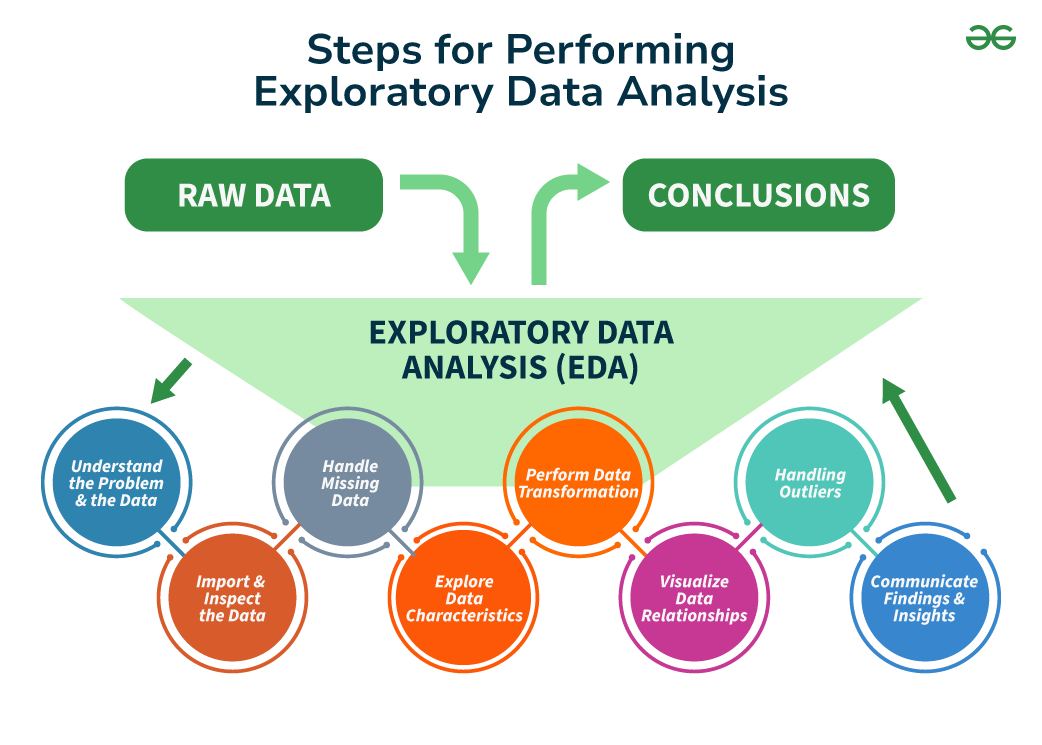
Multivariate analysis examines the relationships between two or more variables in the dataset. It aims to understand how variables interact with one another, which is crucial for most statistical modeling techniques. It include Techniques like[**pair plots**,](https://www.geeksforgeeks.org/python-seaborn-pairplot-method/) which show the relationships between multiple variables at once, helping to see how they interact. Another technique is [**Principal Component Analysis (PCA**](https://www.geeksforgeeks.org/principal-component-analysis-pca/)**)**, which reduces the complexity of large datasets by simplifying them, while keeping the most important information.

In addition to univariate and multivariate analysis, there are specialized EDA techniques tailored for specific types of data or analysis needs:

* [**Spatial Analysis**](https://www.geeksforgeeks.org/what-is-spatial-analysis/): For geographical data, using maps and spatial plotting to understand the geographical distribution of variables.
* **Text Analysis**: Involves techniques like word clouds, frequency distributions, and sentiment analysis to explore text data.
* [**Time Series Analysis**](https://www.geeksforgeeks.org/time-series-data-visualization-in-python/)**:** This type of analysis is mainly applied to statistics sets that have a temporal component. Time collection evaluation entails inspecting and modeling styles, traits, and seasonality inside the statistics through the years. Techniques like line plots, autocorrelation analysis, transferring averages, and [ARIMA](https://www.geeksforgeeks.org/python-arima-model-for-time-series-forecasting/) (AutoRegressive Integrated Moving Average) fashions are generally utilized in time series analysis.

**Steps for Performing Exploratory Data Analysis**

Performing Exploratory Data Analysis (EDA) involves a series of steps designed to help you understand the data you’re working with, uncover underlying patterns, identify anomalies, test hypotheses, and ensure the data is clean and suitable for further analysis.



**Step 1: Understand the Problem and the Data**

The first step in any data analysis project is to clearly understand the problem you’re trying to solve and the data you have. This involves asking key questions such as:

* **What is the business goal or research question**?
* **What are the variables in the data** and what do they represent?
* **What types of data** (numerical, categorical, text, etc.) do you have?
* Are there any known **data quality issues** or **limitations**?
* Are there any **domain-specific concerns** or restrictions?

By thoroughly understanding the problem and the data, you can better plan your analysis, avoid wrong assumptions, and ensure accurate conclusions

**Step 2: Import and Inspect the Data**

After clearly understanding the problem and the data, the next step is to import the data into your analysis environment (like **Python**, **R**, or a spreadsheet tool). At this stage, it’s crucial to examine the data to get an initial understanding of its structure, variable types, and potential issues.

Here’s what you can do:

* **Load the data** into your environment carefully to avoid errors or truncations.
* **Examine the size** of the data (number of rows and columns) to understand its complexity.
* **Check for missing values** and see how they are distributed across variables, since missing data can impact the quality of your analysis.
* **Identify data types** for each variable (like numerical, categorical, etc.), which will help in the next steps of data manipulation and analysis.
* **Look for errors** or inconsistencies, such as invalid values, mismatched units, or outliers, which could signal deeper issues with the data.

By completing these tasks, you’ll be prepared to clean and analyze the data more effectively.

**Step 3: Handle Missing Data**

Missing data is common in many datasets and can significantly affect the quality of your analysis. During **Exploratory Data Analysis (EDA)**, it’s important to identify and handle missing data properly to avoid biased or misleading results.

Here’s how to handle it:

* **Understand the patterns** and possible reasons for missing data. Is it **missing completely at random** (MCAR), **missing at random** (MAR), or **missing not at random** (MNAR)? Knowing this helps decide how to handle the missing data.
* **Decide whether to remove** missing data (listwise deletion) or **impute** (fill in) the missing values. Removing data can lead to biased outcomes, especially if the missing data isn’t MCAR. Imputing values helps preserve data but should be done carefully.
* Use appropriate **imputation methods** like **mean/median imputation**,[**regression**](https://www.geeksforgeeks.org/regression-in-machine-learning/)**imputation**, or machine learning techniques like[**KNN**](https://www.geeksforgeeks.org/k-nearest-neighbours/)or [**decision trees**](https://www.geeksforgeeks.org/decision-tree/)based on the data’s characteristics.
* **Consider the impact** of missing data. Even after imputing, missing data can cause uncertainty and bias, so interpret the results with caution.

Properly handling missing data improves the accuracy of your analysis and prevents misleading conclusions.

**Step 4: Explore Data Characteristics**

After addressing missing data, the next step in **EDA** is to explore the characteristics of your data by examining the **distribution**, [**central tendency**](https://www.geeksforgeeks.org/measures-of-central-tendency/), and **variability** of your variables, as well as identifying any **outliers** or anomalies. This helps in selecting appropriate analysis methods and spotting potential data issues. You should calculate **summary statistics** like **mean**, **median**, **mode**, [**standard deviation**](https://www.geeksforgeeks.org/standard-deviation-formula/), [**skewness**](https://www.geeksforgeeks.org/skewness-measures-and-interpretation/), and [**kurtosis**](https://www.geeksforgeeks.org/how-to-calculate-kurtosis-in-statistics/)for numerical variables. These provide an overview of the data’s distribution and help identify any irregular patterns or issues.

**Step 5: Perform Data Transformation**

Data transformation is an essential step in **EDA** because it prepares your data for accurate analysis and modeling. Depending on your data’s characteristics and analysis needs, you may need to transform it to ensure it’s in the right format.

Common transformation techniques include:

* **Scaling or normalizing** numerical variables (e.g., [**min-max scaling**](https://www.geeksforgeeks.org/data-pre-processing-wit-sklearn-using-standard-and-minmax-scaler/) or **standardization**).
* **Encoding categorical variables** for machine learning (e.g., [**one-hot encoding**](https://www.geeksforgeeks.org/ml-one-hot-encoding/) or **label encoding**).
* Applying **mathematical transformations** (e.g., **logarithmic** or **square root**) to correct skewness or non-linearity.
* **Creating new variables** from existing ones (e.g., calculating ratios or combining variables).
* [**Aggregating or grouping**](https://www.geeksforgeeks.org/grouping-and-aggregating-with-pandas/)data based on specific variables or conditions

**Step 6: Visualize Data Relationship**

Visualization is a powerful tool in the **EDA** process, helping to uncover relationships between variables and identify patterns or trends that may not be obvious from summary statistics alone.

* For categorical variables, create **frequency tables**, **bar plots**, and **pie charts** to understand the distribution of categories and identify imbalances or unusual patterns.
* For numerical variables, generate **histograms**, **box plots**, **violin plots**, and **density plots** to visualize distribution, shape, spread, and potential outliers.
* To explore relationships between variables, use **scatter plots**, **correlation matrices**, or statistical tests like **Pearson’s correlation coefficient** or[**Spearman’s rank correlation**](https://www.geeksforgeeks.org/spearmans-rank-correlation/)

**Step 7: Handling Outliers**

Outliers are data points that significantly differ from the rest of the data, often caused by errors in measurement or data entry. Detecting and handling outliers is important because they can skew your analysis and affect model performance. You can identify outliers using methods like[**interquartile range (IQR)**](https://www.geeksforgeeks.org/interquartile-range-iqr/), [**Z-scores**](https://www.geeksforgeeks.org/z-score-in-statistics/), or **domain-specific rules**. Once identified, outliers can be removed or adjusted depending on the context. Properly managing outliers ensures your analysis is accurate and reliable.

**Step 8: Communicate Findings and Insights**

The final step in **EDA** is to communicate your findings clearly. This involves summarizing your analysis, pointing out key discoveries, and presenting your results in a clear and engaging way.

* Clearly state the **goals** and **scope** of your analysis.
* Provide **context** and background to help others understand your approach.
* Use **visualizations** to support your findings and make them easier to understand.
* Highlight key **insights**, **patterns**, or **anomalies** discovered.
* Mention any **limitations** or challenges faced during the analysis.
* Suggest **next steps** or areas that need further investigation.

Effective conversation is critical for ensuring that your EDA efforts have a meaningful impact and that your insights are understood and acted upon with the aid of stakeholders.

Exploratory Data Analysis (EDA) can be performed using a variety of tools and software, each offering features that deal to different data and analysis needs.

In **Python**, libraries like [**Pandas**](https://www.geeksforgeeks.org/pandas-tutorial/) are essential for data manipulation, providing functions to clean, filter, and transform data. [**Matplotlib**](https://www.geeksforgeeks.org/python-introduction-matplotlib/)is used for creating basic static, interactive, and animated visualizations, while [**Seaborn**](https://www.geeksforgeeks.org/introduction-to-seaborn-python/), built on top of Matplotlib, allows for the creation of more attractive and informative statistical plots. For interactive and advanced visualizations, [**Plotly**](https://www.geeksforgeeks.org/python-plotly-tutorial/) is an excellent choice

In **R**, packages like **ggplot2** are powerful for creating complex and visually appealing plots from data frames. **dplyr** helps in data manipulation, making tasks like filtering and summarizing easier, and **tidyr** ensures your data is in a tidy format, making it easier to work with.