I. The Significance of Exploratory Data Analysis (EDA)

Data is accumulated and stored across a wide range of disciplines—including science, economics, engineering, and marketing—primarily in electronic databases. To make appropriate and well-informed decisions, this data must be analyzed effectively. However, when dealing with datasets containing more than just a few data points, it becomes practically impossible to extract meaningful insights without the aid of computer programs.

This is where data mining comes into play—through a series of distinct analytical processes aimed at uncovering valuable information. Exploratory Data Analysis (EDA) is a key—and often the first—step in data mining.

Why EDA Matters

- Reveals the "ground truth":

  EDA allows us to understand what the data actually says without imposing prior assumptions. This unbiased exploration forms the foundation for all subsequent analysis.

- Facilitates hypothesis generation:

  Rather than testing preconceived theories, EDA helps data scientists discover hidden patterns, spot anomalies, and formulate new hypotheses that can guide future data collection or experimental design.

- Supports data-driven decision-making:

  By summarizing and visualizing data, EDA provides actionable insights that stakeholders can use for strategic business intelligence.

- Guides modeling choices:

  Understanding the structure, quality, and relationships within the data helps determine which models or algorithms are most suitable for the problem at hand.

Core Components of EDA

EDA integrates three essential elements:

1. Summarizing data – using descriptive statistics to capture central tendencies, spread, and shape.
2. Statistical analysis – identifying distributions, correlations, outliers, and data quality issues.
3. Data visualization – employing graphs and charts (e.g., histograms, scatter plots, box plots) to reveal trends and anomalies intuitively.

Tools Enabling EDA

Modern open-source tools make EDA both powerful and accessible:

- Python: A leading programming language in data science.

- pandas for data summarization and manipulation
- SciPy for statistical analysis
- Matplotlib and Plotly for data visualization

These tools empower analysts to explore, understand, and communicate insights effectively.

II. Steps in EDA

1. Problem Definition

Before attempting to extract useful insights from the data, it is essential to clearly define the business problem to be solved. The problem definition acts as the driving force for executing a data analysis plan.

Main tasks involved include:

- Defining the main objective of the analysis
- Defining the main deliverables
- Outlining roles and responsibilities
- Obtaining the current status of the data
- Defining the timetable
- Performing a cost/benefit analysis: Based on this problem definition, an execution plan can be formulated.

2. Data Preparation

This step involves preparing the dataset for actual analysis. Key activities include:

- Identifying data sources
- Defining data schemas and tables
- Understanding the main characteristics of the data
- Cleaning the dataset
- Removing irrelevant data
- Transforming the data
- Dividing the data into required chunks for analysis

3. Data Analysis

This is one of the most crucial steps, focusing on descriptive statistics and deeper data examination. Main tasks include:

- Summarizing the data
- Identifying hidden correlations and relationships
- Developing predictive models

- o Evaluating the models
- o Calculating model accuracies

Techniques used for data summarization include: summary tables, graphs, descriptive statistics, inferential statistics, correlation analysis, searching, grouping, and mathematical models.

4. Development and Representation of the Results

This step involves presenting the analyzed data to the target audience using visual and tabular formats such as:

- o Graphs
- o Summary tables
- o Maps
- o Diagrams

This step is essential because the results must be interpretable by business stakeholders—a core goal of EDA.

Common graphical techniques include: scatter plots, character plots, histograms, box plots, residual plots, mean plots, and others. (Several visualization types are explored in Chapter 2: Visual Aids for EDA.)

III. Making Sense of Data – Numerical and Categorical

It is crucial to identify the type of data under analysis. In this section, we are going to learn about different types of data that you can encounter during analysis. Different disciplines store different kinds of data for different purposes. For example:

- Medical researchers store patients' data
- Universities store students' and teachers' data
- Real estate industries store house and building datasets

A dataset contains many observations about a particular object. For instance, a dataset about patients in a hospital can contain multiple observations. A patient can be described by attributes such as:

- Patient Identifier (ID)
- Name
- Address
- Weight
- Date of Birth
- Email

- Gender

Each of these attributes is a variable, and each observation holds a specific value for each variable. For example:
PATIENT_ID = 1001
Name = Yoshmi Mukhiya
Address = Mannsverk 61, 5094, Bergen, Norway
Date of Birth = 10th July 2018
Email = yoshmimukhiya@gmail.com
Weight = 10
Gender = Female
Such data is typically stored in a Database Management System (DBMS) in the form of tables or schemas. An example of a patient information table is shown below:

PATIENT ID	NAME	ADDRESS	DOB	EMAIL	GENDER	WEIGHT
001	Suresh Kumar Mukhiya	Mannsverk, 61	30.12.1989	skmu@hvl.no	Male	68
002	Yoshmi Mukhiya	Mannsverk 61, 5094, Bergen	10.07.2018	yoshmimukhiya@gmail.com	Female	—
003	Anju Mukhiya	Mannsverk 61, 5094, Bergen	10.12.1997	anjumukhiya@gmail.com	Female	24
004	Asha Gaire	Butwal, Nepal	30.11.1990	aasha.gaire@gmail.com	Female	23
005	Ola Nordmann	Danmark, Sweden	12.12.1789	ola@gmail.com	Male	75

Note: The table above contains five observations (rows 001–005). Each observation includes values for the same set of variables (Patient ID, Name, Address, DOB, Email, Gender, Weight).
Most datasets broadly fall into two main categories:
1. Numerical Data (Quantitative)

This type of data involves measurable quantities, such as:

- Age
- Height
- Weight
- Blood pressure
- Heart rate
- Temperature

- Number of teeth or family members

Numerical data is further divided into:
a) Discrete Data

- Countable values with finite possibilities.
- Examples:
  - Number of heads in 200 coin flips (values from 0 to 200)
  - Country (Nepal, India, Norway, Japan — fixed categories represented numerically)
  - Student rank in a class (1st, 2nd, 3rd, etc.)

A discrete variable takes a fixed number of distinct values.

b) Continuous Data

- Can take any value within a range (infinite possibilities).
- Examples:
  - Temperature of a city today
  - Weight (as in the patient table above)

A continuous variable represents continuous data.

Continuous data typically follows either an interval or ratio scale (discussed in the Measurement Scales section).

2. Categorical Data (Qualitative)

This type captures qualities or characteristics, such as:

- Gender
- Marital status
- Type of address
- Movie genres

Common examples include:

- Gender: Male, Female, Other, Unknown
- Marital Status: Annulled, Divorced, Married, Widowed, etc.
- Movie Genres: Action, Comedy, Drama, Sci-Fi, etc.
- Blood Type: A, B, AB, O
- Types of Drugs: Stimulants, Depressants, Hallucinogens, etc.

A categorical variable has a limited number of possible values and is often treated as an enumerated type in programming.

Types of Categorical Variables:

- Binary (Dichotomous): Exactly two categories
  - Example: Success/Failure, Yes/No
- Polytomous: More than two categories
  - Example: Marital status with multiple options

Most categorical data follows either a nominal or ordinal measurement scale (explained in the next section).

IV. Measurement Scales

In statistics and data analysis, measurement scales (also called levels of measurement) describe the nature of information contained within the values assigned to variables. Understanding the type of measurement scale associated with your data is critical because it dictates what kinds of mathematical operations, statistical techniques, and visualizations are valid.

There are four fundamental types of measurement scales:

1. Nominal Scale
2. Ordinal Scale
3. Interval Scale
4. Ratio Scale

These scales form a hierarchy—from the least informative (nominal) to the most informative (ratio)—and each builds upon the properties of the one before it.

1. Nominal Scale

The nominal scale is used for labeling or categorizing data without any inherent order or numerical value. The word “nominal” comes from the Latin word nomen, meaning "name." In this scale, numbers or labels serve only as identifiers, not as quantities.

Key Properties

- No natural order among categories.
- Arithmetic operations are meaningless (you cannot add, subtract, or average categories).
- Only equality and inequality (= or ≠) are meaningful.

Statistical Tools Allowed

- Frequency counts
- Mode (most frequent category)

- Chi-square tests (for independence or goodness-of-fit)

Examples

- Gender: Male, Female, Non-binary, Prefer not to say
- Blood types: A, B, AB, O
- Country of residence: Nepal, Japan, Brazil, Canada
- Languages spoken: English, Mandarin, Spanish, Swahili
- Vehicle brands: Toyota, Ford, BMW, Tesla

2. Ordinal Scale

The ordinal scale not only categorizes data (like nominal) but also ranks the categories in a meaningful order. However, the differences between ranks are not necessarily equal or quantifiable.

Key Properties

- Order matters (e.g., 1st > 2nd > 3rd).
- Distances between values are unknown or inconsistent.
- You can say "A is better than B," but not "A is twice as good as B."

Statistical Tools Allowed

- Median (middle value in ordered list)
- Percentiles and quartiles
- Non-parametric tests (e.g., Mann-Whitney U test, Kruskal-Wallis test)

Mean is generally not appropriate because intervals between points aren't uniform.

Examples

- Customer satisfaction:
    - o Very Dissatisfied
    - o Dissatisfied
    - o Neutral
    - o Satisfied
    - o Very Satisfied
- Education level:
    - o Elementary
    - o High School
    - o Bachelor's

  - o Master’s
  - o PhD
- Socioeconomic status: Low, Middle, High

3. Interval Scale

The interval scale has ordered categories with equal intervals between values, but no true zero point. This means ratios are not meaningful, even though differences are.

Key Properties

- Order + equal spacing between values
- Zero does not mean “absence” of the quantity
- Addition and subtraction are valid, but multiplication and division are not

Statistical Tools Allowed

- Mean, median, mode
- Standard deviation, variance
- Parametric tests (e.g., t-tests, ANOVA, Pearson correlation)

Examples

- Temperature in Celsius or Fahrenheit:
  - o 20°C is not twice as hot as 10°C
  - o But the difference between 20°C and 10°C is the same as between 30°C and 20°C
  - o 0°C ≠ “no temperature” — it’s just the freezing point of water
- Calendar years:
  - o Year 2000 is not “twice” year 1000
  - o But the gap between 1990 and 2000 is the same as between 2000 and 2010
- IQ scores:
  - o An IQ of 100 vs. 150 — difference is meaningful
  - o But 150 is not 1.5× “smarter” than 100 in any scientifically absolute sense

4. Ratio Scale

The ratio scale is the most informative measurement level. It possesses all the properties of the interval scale plus a true zero point, which represents complete absence of the measured attribute. This allows meaningful ratios.

Key Properties

- Order + equal intervals + true zero
- All arithmetic operations are valid: +, −, ×, ÷
- Ratios are interpretable: e.g., “10 kg is twice as heavy as 5 kg”

Statistical Tools Allowed

- All descriptive and inferential statistics
- Geometric mean, coefficient of variation
- Advanced modeling (regression, machine learning)

Examples

- Height: 180 cm is 1.5× taller than 120 cm
- Weight: 80 kg vs. 40 kg   twice as heavy
- Age: A 40-year-old is twice as old as a 20-year-old
- Income: $60,000/year vs. $30,000/year   double the income
- Time duration: 10 seconds vs. 5 seconds   2× longer
- Distance: 100 km vs. 25 km   4× farther

V. Comparing EDA with classical and Bayesian Analysis

There are several approaches to data analysis. The most relevant ones include:

- Classical Data Analysis

In the classical approach, the process typically follows a fixed sequence:

Problem Definition   Data Collection   Model Development   Data Analysis   Results Communication.

Here, a model (deterministic or probabilistic) is imposed before examining the data in detail. The analysis is then conducted under the assumptions of that pre-specified model.

- Exploratory Data Analysis (EDA) Approach

EDA follows a similar sequence to classical analysis, except that the steps of model development and data analysis are swapped. The main focus is on the data itself—its structure, patterns, anomalies, and relationships—before imposing any formal model. In EDA, we generally do not assume or impose deterministic or probabilistic models upfront. Instead, we let the data guide hypothesis generation and modeling decisions.

- Bayesian Data Analysis Approach

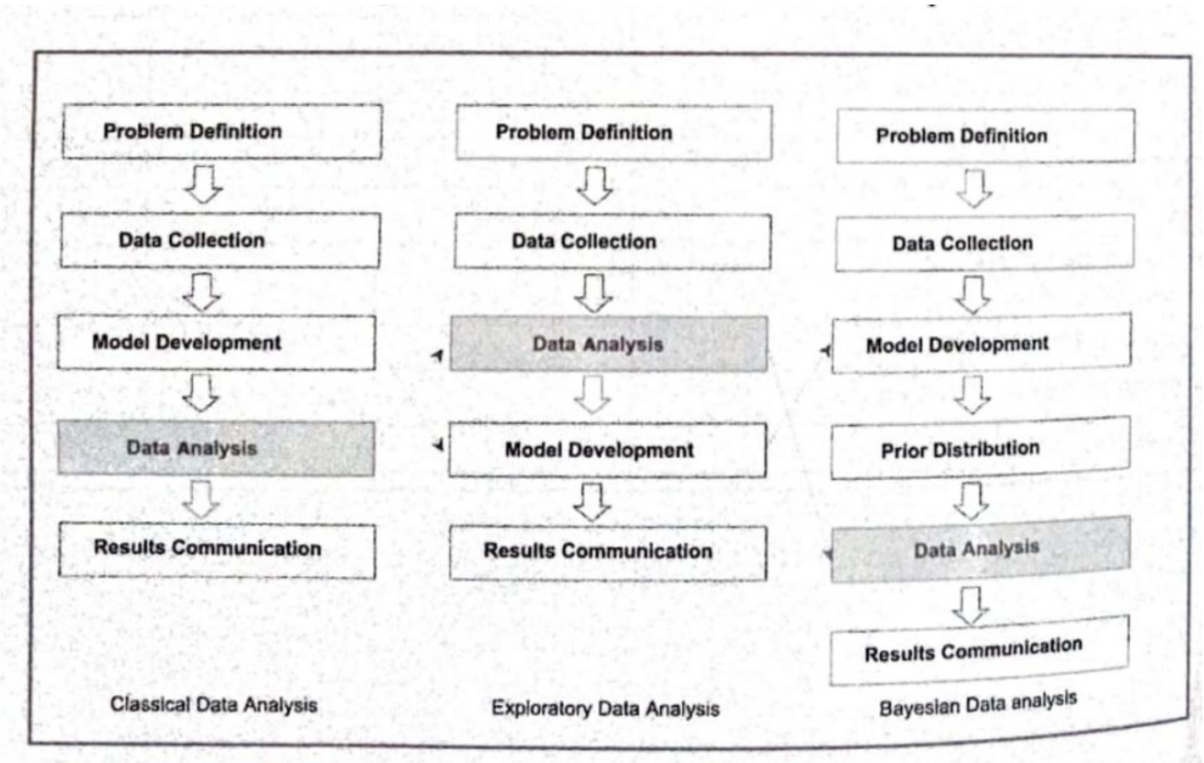
The Bayesian approach incorporates prior probability distributions into the analysis process. A prior distribution represents existing beliefs or knowledge about a parameter before observing the current data. The workflow integrates this prior knowledge with observed data to produce a posterior distribution. The typical sequence is:

Problem Definition Data Collection Model Development Prior Distribution Data Analysis Results Communication.

The following diagram illustrates the differences in the execution steps across the three approaches:

Classical Data Analysis	Exploratory Data Analysis (EDA)	Bayesian Data Analysis
Problem Definition	Problem Definition	Problem Definition
Data Collection	Data Collection	Data Collection
Model Development	Data Analysis	Model Development
Data Analysis	Model Development	Prior Distribution
Results Communication	Results Communication	Data Analysis
		Results Communication

In practice, data analysts and data scientists often combine elements from all three approaches to extract meaningful insights. It is rarely productive—or even possible—to declare one approach universally "best." Each paradigm has its own strengths and is suited to different types of problems and data contexts.



VI. Software tools available for EDA

Several software tools are available to facilitate Exploratory Data Analysis (EDA). Below are some of the widely used open-source tools:

- Python:

  An open-source programming language extensively used in data analysis, data mining, and data science.

  Website: https://www.python.org/

- R Programming Language:

  An open-source language specifically designed for statistical computing and graphical data analysis.

  Website: https://www.r-project.org

- Weka:

  An open-source data mining package that includes a variety of EDA tools and machine learning algorithms.

  Website: https://www.cs.waikato.ac.nz/ml/weka/

- KNIME:

  An open-source data analytics platform built on the Eclipse framework, supporting visual programming for data workflows.

Website: https://www.knime.com/

Getting Started with EDA – Numpy – pandas – Scipy – Matplotlib

As mentioned earlier, we are going to use Python as the main tool for data analysis. Python has been consistently ranked among the top 10 programming languages and is widely adopted for data analysis and data mining by data science experts.

NumPy

- Create arrays with NumPy, copy arrays, and divide
- Perform different operations on NumPy arrays
- Understand array selections, advanced indexing, and expanding
- Work with multi-dimensional arrays
- Use linear algebraic functions and built-in NumPy utilities

Pandas

- Understand and create DataFrame objects
- Subset and index data
- Apply arithmetic functions and mapping with pandas
- Manage indexes
- Build styles for visual analysis

Matplotlib

- Load linear datasets
- Adjust axes, grids, labels, titles, and legends
- Save plots

SciPy

- Import the package
- Use statistical modules from SciPy
- Perform descriptive statistics
- Conduct inference and data analysis

1. NumPy

NumPy is essential for numerical computing in Python. Below are basic NumPy operations for EDA:

- Importing and Creating Arrays

```
import numpy as np

# 1D array
my1DArray = np.array([1, 8, 27, 64])
print(my1DArray)

# 2D array
my2DArray = np.array([[1, 2, 3, 4], [2, 4, 9, 16], [4, 8, 18, 32]])
print(my2DArray)

# 3D array
my3DArray = np.array([[[1, 2, 3, 4], [5, 6, 7, 8]], [[1, 2, 3, 44], [9, 10, 11, 12]]])
print(my3DArray)
```

- Array Attributes

```
print(my2DArray.data)      # Memory address
print(my2DArray.shape)     # Shape
print(my2DArray.dtype)     # Data type
print(my2DArray.strides)   # Strides
```

- Built-in Array Creation Functions

```
np.ones((3, 4))                   # Array of ones
np.zeros((2, 3, 4), dtype=np.int16) # Array of zeros
np.random.random((2, 2))          # Random values
np.empty((3, 2))                  # Uninitialized array
np.full((2, 2), 7)                # Fill with value
np.arange(10, 25, 5)              # Evenly spaced values (step)
np.linspace(0, 2, 9)              # Evenly spaced values (count)
```

- Array Inspection

```
print(my2DArray.ndim)     # Dimensions
print(my2DArray.size)     # Total elements
print(my2DArray.flags)    # Memory layout info
print(my2DArray.itemsize) # Bytes per element
print(my2DArray.nbytes)   # Total bytes consumed
```

- Basic Mathematical Operations

```
X = np.array([[1, 2, 3], [2, 3, 4]])
Y = np.array([[1, 4, 9], [2, 3, -2]])

np.add(X, Y)       # Addition
np.subtract(X, Y)  # Subtraction
np.multiply(X, Y)  # Element-wise multiplication
np.divide(X, Y)    # Division
np.remainder(X, Y)  # Remainder
```

- Indexing and Slicing

```
x = np.array([10, 20, 30, 40, 50])
print(x[0:2])             # First two elements

y = np.array([[1, 2, 3, 4], [9, 10, 11, 12]])
print(y[0:2, 1])          # Column 1 from rows 0–1
print(y[y >= 2])          # Conditional selection
```

2. Pandas

Pandas is a powerful library for data manipulation and analysis, developed by Wes McKinney.
It has two components:

i. Series(1-dimensional)
ii. DataFrame(2-dimensional)
Setup and Basics

```
import numpy as np
import pandas as pd
print("Pandas Version:", pd.__version__)
```

Creating DataFrames

- From Series:

```
import pandas as pd
s=pd.Series([1,2,3,4,5])
print(s)
```

- From Dictionary:

```
df = [{'A': 'Aeroplane', 'B': 'Bat', 'C': 'Cat'}]
f = pd.DataFrame(df)
```

- DataFrame Inspection

```
df.info()  # Shows rows, columns, dtypes, memory usage
```

- Row and Column Selection

```
df.iloc[10]        # Single row
df.iloc[0:10]      # First 10 rows
df.iloc[10:15]     # Rows 10–14
df.iloc[-2:]       # Last 2 rows
df.iloc[::2, 3:5]  # Every other row, columns 3–4
```

3. SciPy

SciPy is an open-source scientific computing library that builds on NumPy. It provides advanced functions for optimization, integration, interpolation, eigenvalue problems, and statistical analysis.

- Key module for EDA: scipy.stats
- Used for probability distributions, statistical tests, and descriptive statistics
- Will be explored in depth in upcoming chapters

4. Matplotlib

Matplotlib is the foundational plotting library in Python, offering:

- A wide variety of customizable 2D plots (line, scatter, bar, histogram, etc.)
- Support for professional reporting, dashboards, web apps, and interactive visualizations
- Full control over every plot element (axes, labels, legends, grids, etc.)