UNIT I:

What is Data Analysis? Differences between Data Analysis and Analytics, What is Python, Why Python for Data Analysis? What is Library, Essential Python Libraries. Python Language basics, I-Python and Jupyter Notebook. Python Language Basics.

**Differences Between Data Analysis and Data Analytics**

| **Aspect** | **Data Analysis** | **Data Analytics** |
| --- | --- | --- |
| **Definition** | Focuses on extracting insights from existing data. | Involves predictive models and forecasting. |
| **Scope** | Descriptive analysis of historical data. | Predictive or prescriptive analysis. |
| **Approach** | Answering "What happened?" | Answering "What will happen?" or "What to do?" |
| **Tools** | Excel, SQL, Python | Python, R, Power BI, Machine Learning |

**What is Data Analysis?**

**Data Analysis** refers to the process of inspecting, cleaning, transforming, and modeling data to discover useful information, support decision-making, and draw conclusions.

**Steps in Data Analysis:**

1. **Data Collection** – Gathering raw data from different sources.
2. **Cleaning** – Handling missing data and inconsistencies.
3. **Exploration** – Visualizing the data and understanding patterns.
4. **Modeling** – Building predictive models, if needed.
5. **Interpretation** – Drawing meaningful conclusions.

**Differences Between Data Analysis and Data Analytics**

| **Aspect** | **Data Analysis** | **Data Analytics** |
| --- | --- | --- |
| **Definition** | Focuses on extracting insights from existing data. | Involves predictive models and forecasting. |
| **Scope** | Descriptive analysis of historical data. | Predictive or prescriptive analysis. |
| **Approach** | Answering "What happened?" | Answering "What will happen?" or "What to do?" |
| **Tools** | Excel, SQL, Python | Python, R, Power BI, Machine Learning |

**What is Python?**

**Python** is a high-level, interpreted programming language known for its simplicity and readability. It is widely used in various fields like web development, data science, machine learning, automation, and more.

**Features of Python:**

1. Easy to learn and use.
2. Interpreted, so no need to compile code.
3. Strong support for libraries and frameworks.
4. Open-source and community-supported.

**Why Python for Data Analysis?**

Python is preferred for data analysis because:

1. **Ease of Use**: Its syntax is easy to learn.
2. **Powerful Libraries**: Libraries like NumPy, Pandas, and Matplotlib simplify data tasks.
3. **Scalable and Fast**: Python can handle large datasets efficiently.
4. **Integration**: It integrates with databases, cloud tools, and web services.
5. **Community Support**: A large community ensures quick problem resolution.

**What is a Library?**

A **library** is a collection of pre-written code that provides specific functionality, such as numerical operations, visualization, or data manipulation. It allows developers to reuse code instead of writing it from scratch.

**Essential Python Libraries for Data Analysis**

1. **NumPy**: Used for numerical computations.
2. **Pandas**: Used for data manipulation and analysis.
3. **Matplotlib**: Used for data visualization.
4. **Seaborn**: An extension of Matplotlib for statistical plots.
5. **SciPy**: Used for scientific computations.
6. **Scikit-learn**: Used for machine learning.

**Python Language Basics with Example**

**Variables and Data Types**

| # Variables and Data Types name = "Alice" age = 25 height = 5.4 is\_student = True   print(f"Name: {name}, Age: {age}, Height: {height}, Student: {is\_student}") |
| --- |

**Control Structures: if-else, loops**

| # If-else Example score = 85 if score >= 60:  print("Pass") else:  print("Fail")   # Loop Example |
| --- |

for i in range(5):

print(f"Number: {i}")

**Functions in Python**

| # Function Example def greet(name):  return f"Hello, {name}!"   print(greet("Alice")) |
| --- |

**IPython and Jupyter Notebook**

**What is IPython?**

IPython is an enhanced Python shell with additional functionality like interactive computing and easy debugging. It offers features like tab completion and inline plotting.

**What is a Jupyter Notebook?**

**Jupyter Notebook** is an open-source web-based tool that allows users to create and share documents with live code, visualizations, and narrative text. It is widely used for data science and machine learning projects.

**Installing Jupyter Notebook:**

| !pip install notebook |
| --- |

**Example Program for Data Analysis Using Python**

Pandas for data manipulation and Matplotlib for visualization.

| # Importing Essential Libraries import pandas as pd import matplotlib.pyplot as plt   # Creating a DataFrame data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],  'Age': [24, 27, 22, 32],  'Salary': [85, 78, 90, 88]}   df = pd.DataFrame(data)   # Display the DataFrame print(df)   # Basic Analysis print("\nAverage Age:", df['Age'].mean()) print("Highest salary:", df['Salary'].max())   # Plotting the Scores plt.bar(df['Name'], df['Score'], color='skyblue') plt.xlabel('Name') plt.ylabel('Score') plt.title('Salary') plt.show() |
| --- |