www.aiquest.org

PANDAS

শেখার কোন শেষ নাই!

5 Mins to Pandas

Instructor:

Rashedul Alam Shakil
Founder of aiQuest Intelligence
M.Sc. in Data Science at FAU Germany

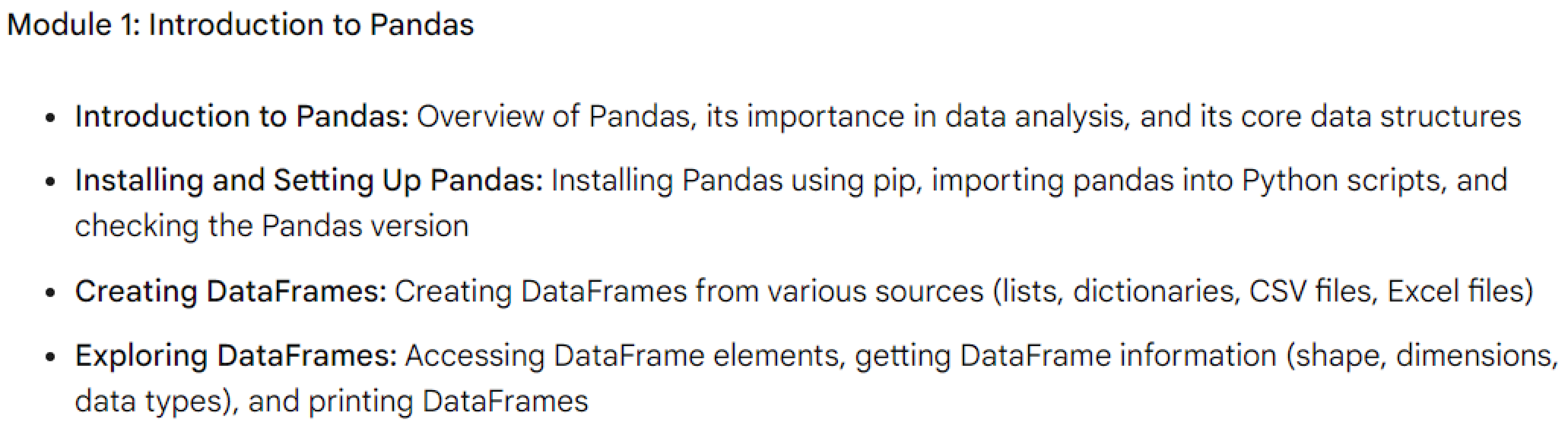
5 Minutes to Pandas

Course Planning

Module 1: Introduction to Pandas
Module 2: Data Manipulation with Pandas
Module 3: Data Cleaning and Preparation with Pandas
Module 4: Data Analysis with Pandas
Module 5: Data Visualization with Pandas
Module 6: Advanced Pandas Topics
Module 7: Pandas Profiling
Module 8: Integrating SQL with Pandas
Module 9: Upcoming Functions

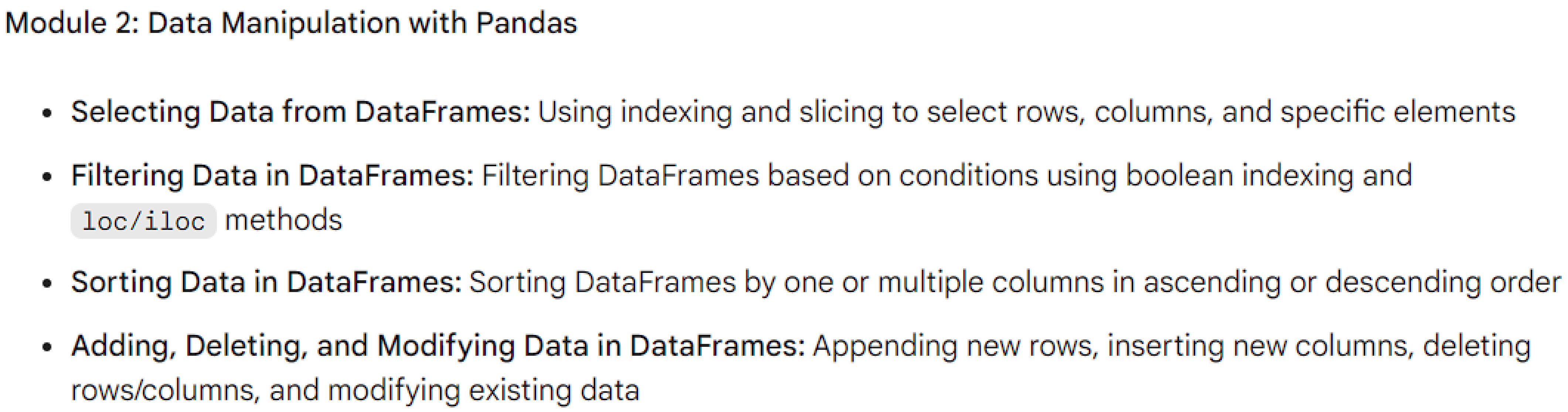
5 Minutes to Pandas

Course Planning



Module 1: Introduction to Pandas

- **Introduction to Pandas:** Overview of Pandas, its importance in data analysis, and its core data structures
- **Installing and Setting Up Pandas:** Installing Pandas using pip, importing pandas into Python scripts, and checking the Pandas version
- **Creating DataFrames:** Creating DataFrames from various sources (lists, dictionaries, CSV files, Excel files)
- **Exploring DataFrames:** Accessing DataFrame elements, getting DataFrame information (shape, dimensions, data types), and printing DataFrames

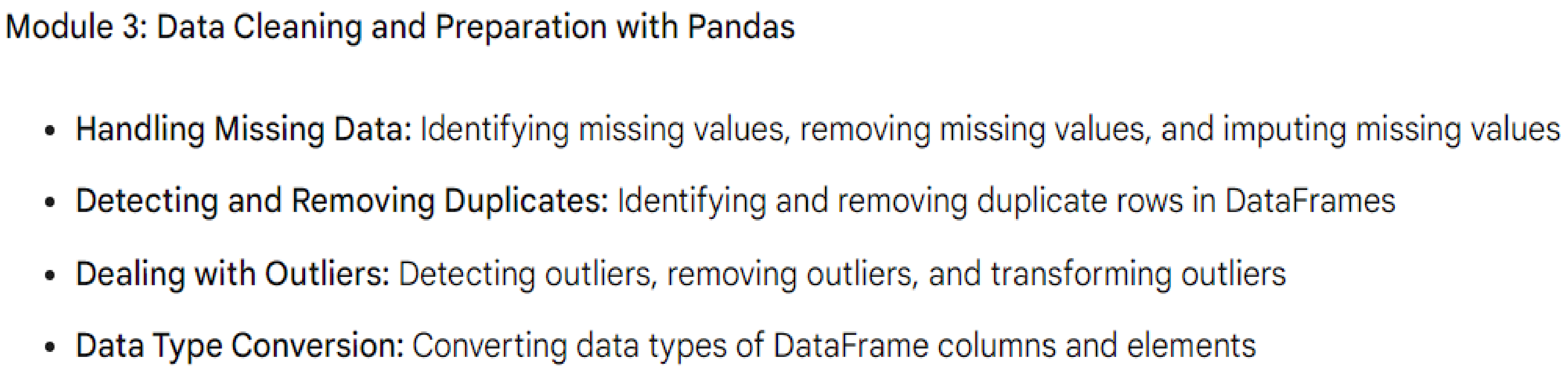


Module 2: Data Manipulation with Pandas

- **Selecting Data from DataFrames:** Using indexing and slicing to select rows, columns, and specific elements
- **Filtering Data in DataFrames:** Filtering DataFrames based on conditions using boolean indexing and `loc/iloc` methods
- **Sorting Data in DataFrames:** Sorting DataFrames by one or multiple columns in ascending or descending order
- **Adding, Deleting, and Modifying Data in DataFrames:** Appending new rows, inserting new columns, deleting rows/columns, and modifying existing data

5 Minutes to Pandas

Course Planning

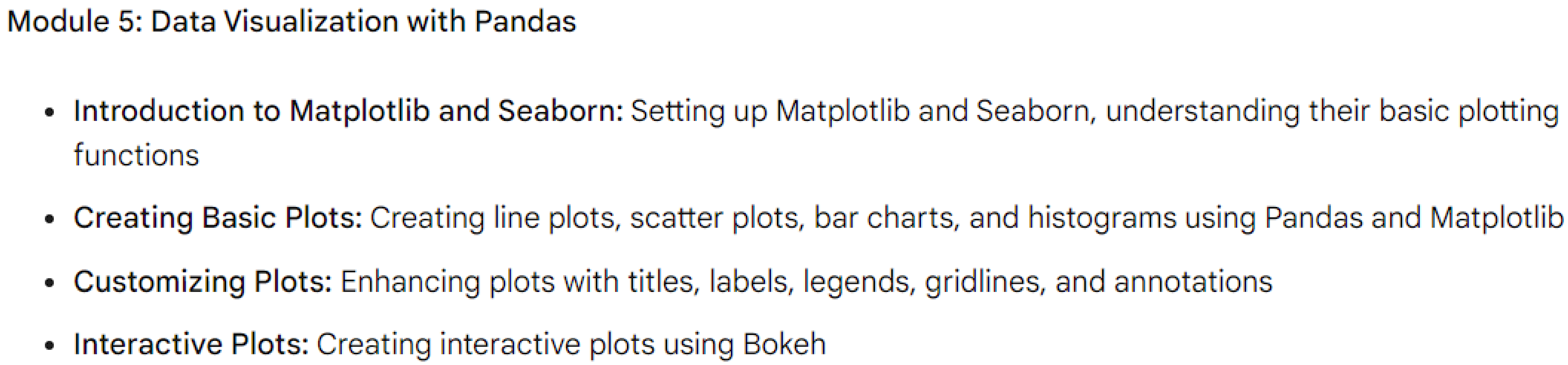


Module 3: Data Cleaning and Preparation with Pandas

- **Handling Missing Data:** Identifying missing values, removing missing values, and imputing missing values
- **Detecting and Removing Duplicates:** Identifying and removing duplicate rows in DataFrames
- **Dealing with Outliers:** Detecting outliers, removing outliers, and transforming outliers
- **Data Type Conversion:** Converting data types of DataFrame columns and elements

5 Minutes to Pandas

Course Planning

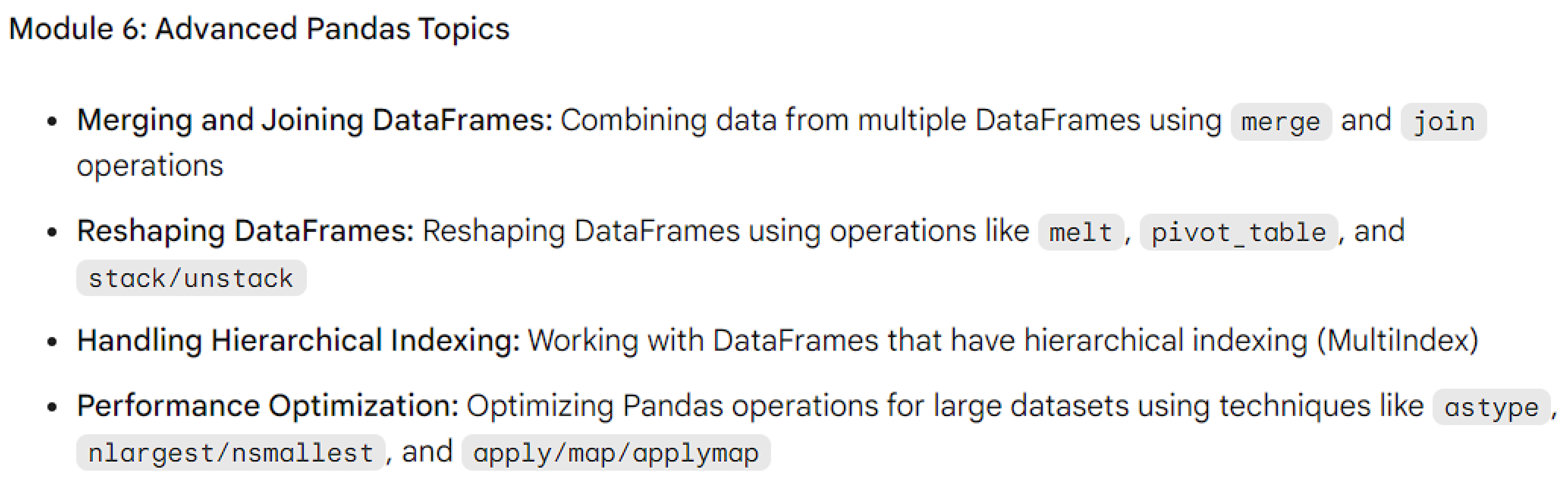


Module 5: Data Visualization with Pandas

- **Introduction to Matplotlib and Seaborn:** Setting up Matplotlib and Seaborn, understanding their basic plotting functions
- **Creating Basic Plots:** Creating line plots, scatter plots, bar charts, and histograms using Pandas and Matplotlib
- **Customizing Plots:** Enhancing plots with titles, labels, legends, gridlines, and annotations
- **Interactive Plots:** Creating interactive plots using Bokeh

5 Minutes to Pandas

Course Planning

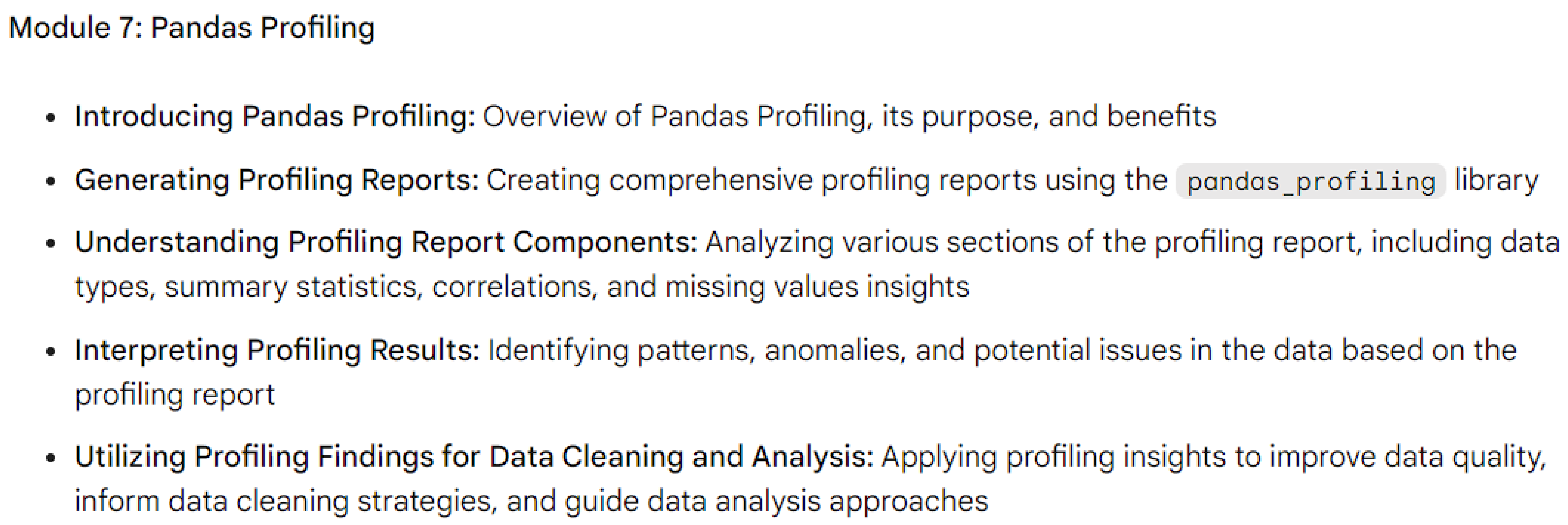


Module 6: Advanced Pandas Topics

- **Merging and Joining DataFrames:** Combining data from multiple DataFrames using `merge` and `join` operations
- **Reshaping DataFrames:** Reshaping DataFrames using operations like `melt`, `pivot_table`, and `stack/unstack`
- **Handling Hierarchical Indexing:** Working with DataFrames that have hierarchical indexing (MultiIndex)
- **Performance Optimization:** Optimizing Pandas operations for large datasets using techniques like `astype`, `nlargest/nsmallest`, and `apply/map/applymap`

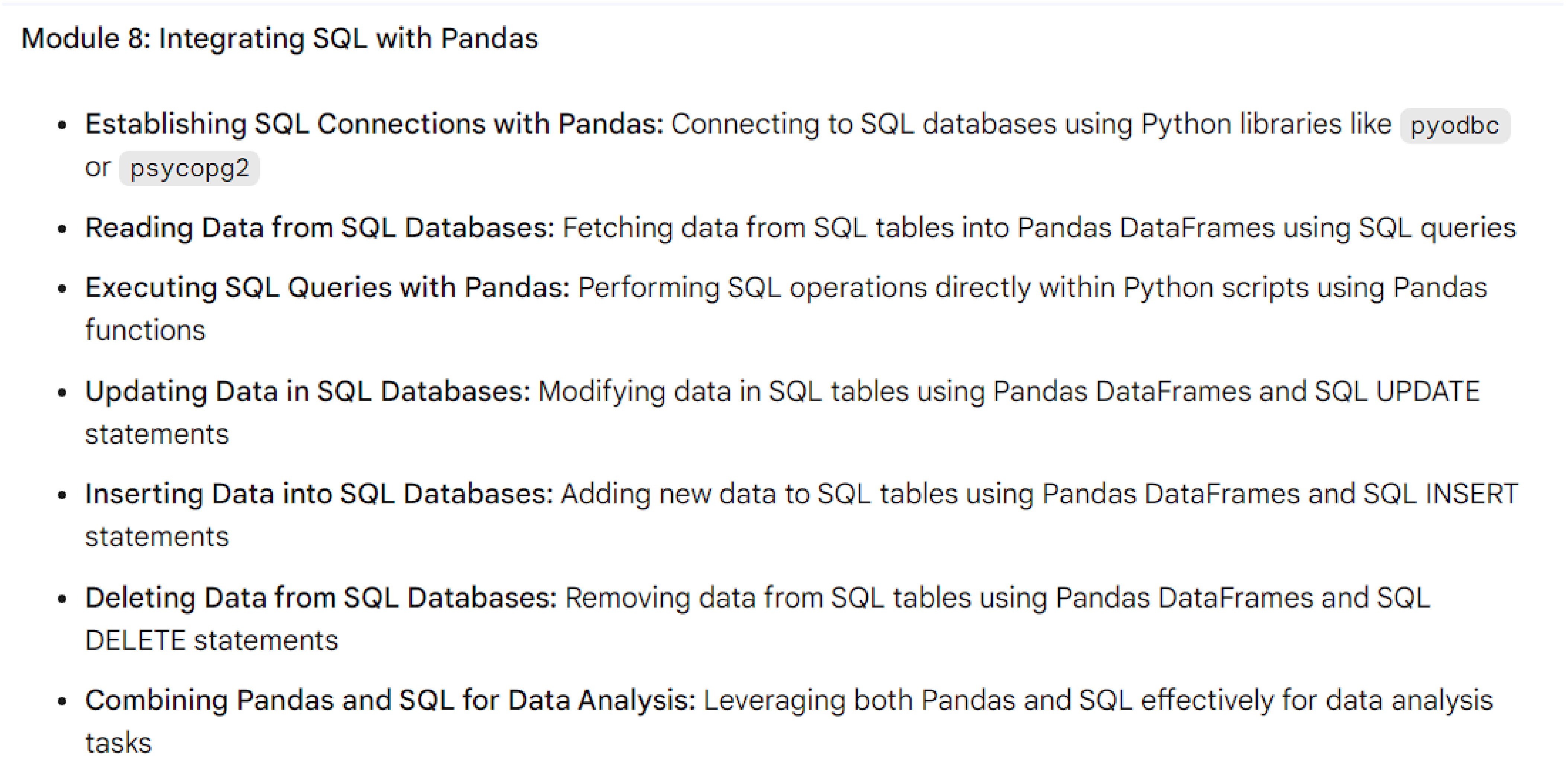
5 Minutes to Pandas

Course Planning



Module 7: Pandas Profiling

- **Introducing Pandas Profiling:** Overview of Pandas Profiling, its purpose, and benefits
- **Generating Profiling Reports:** Creating comprehensive profiling reports using the `pandas_profiling` library
- **Understanding Profiling Report Components:** Analyzing various sections of the profiling report, including data types, summary statistics, correlations, and missing values insights
- **Interpreting Profiling Results:** Identifying patterns, anomalies, and potential issues in the data based on the profiling report
- **Utilizing Profiling Findings for Data Cleaning and Analysis:** Applying profiling insights to improve data quality, inform data cleaning strategies, and guide data analysis approaches



Module 8: Integrating SQL with Pandas

- **Establishing SQL Connections with Pandas:** Connecting to SQL databases using Python libraries like `pyodbc` or `psycopg2`
- **Reading Data from SQL Databases:** Fetching data from SQL tables into Pandas DataFrames using SQL queries
- **Executing SQL Queries with Pandas:** Performing SQL operations directly within Python scripts using Pandas functions
- **Updating Data in SQL Databases:** Modifying data in SQL tables using Pandas DataFrames and SQL UPDATE statements
- **Inserting Data into SQL Databases:** Adding new data to SQL tables using Pandas DataFrames and SQL INSERT statements
- **Deleting Data from SQL Databases:** Removing data from SQL tables using Pandas DataFrames and SQL DELETE statements
- **Combining Pandas and SQL for Data Analysis:** Leveraging both Pandas and SQL effectively for data analysis tasks

Continue…..

www.aiquest.org

PANDAS

শেখার কোন শেষ নাই!

5 Mins to Pandas

Instructor:

Rashedul Alam Shakil
Founder of aiQuest Intelligence
M.Sc. in Data Science at FAU Germany

Pandas Basics

What is Pandas?

Pandas is a powerful, open-source Python library that provides high-performance, easy-to-use data structures and data analysis tools. It’s a must-have library for data scientists and analysts as it makes working with data easier and more efficient.

Fig Source: Google.com

Pandas Basics

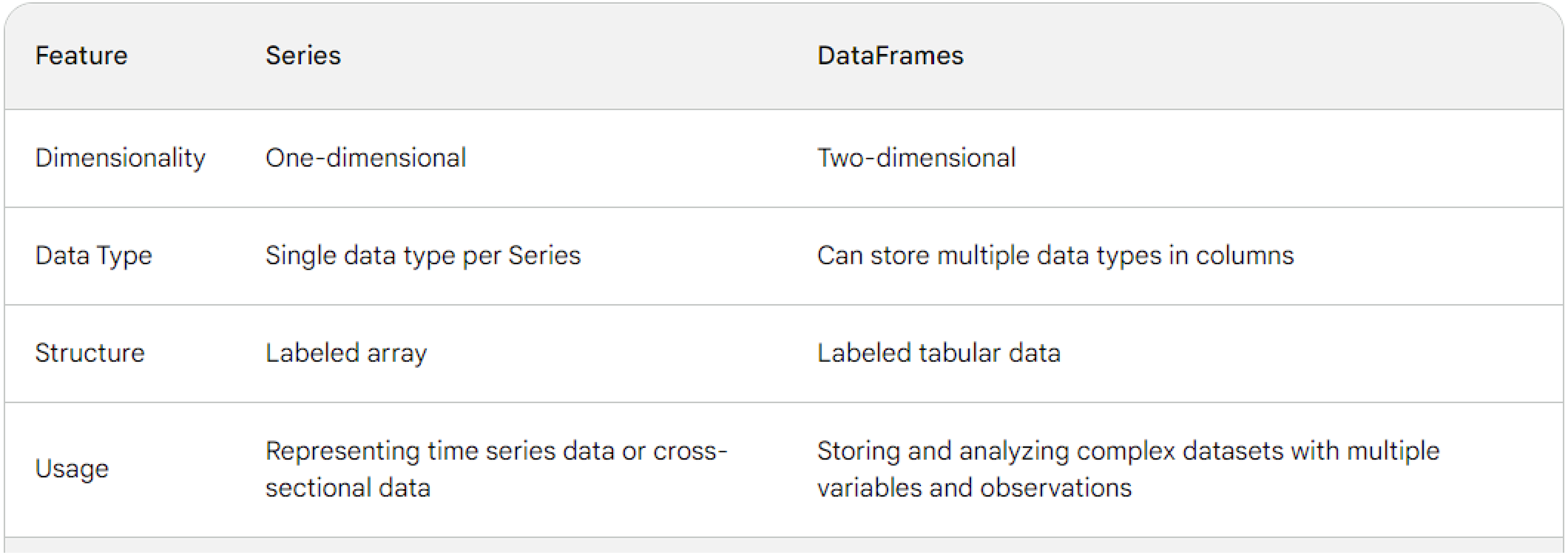
Data Structure in Pandas

- Series: A Series is a one-dimensional labeled array capable of holding a variety of data types, including integers, floats, strings, and booleans. It is similar to a NumPy array, but with the addition of labels for the data points. Series are commonly used to represent time series data or cross-sectional data.

- DataFrames: A DataFrame is a two-dimensional labeled tabular data structure with rows and columns. It is similar to a spreadsheet or database table. DataFrames can store a variety of data types in its columns, and each row represents a distinct observation. DataFrames are well-suited for storing and analyzing complex datasets with multiple variables and observations.

Pandas Basics

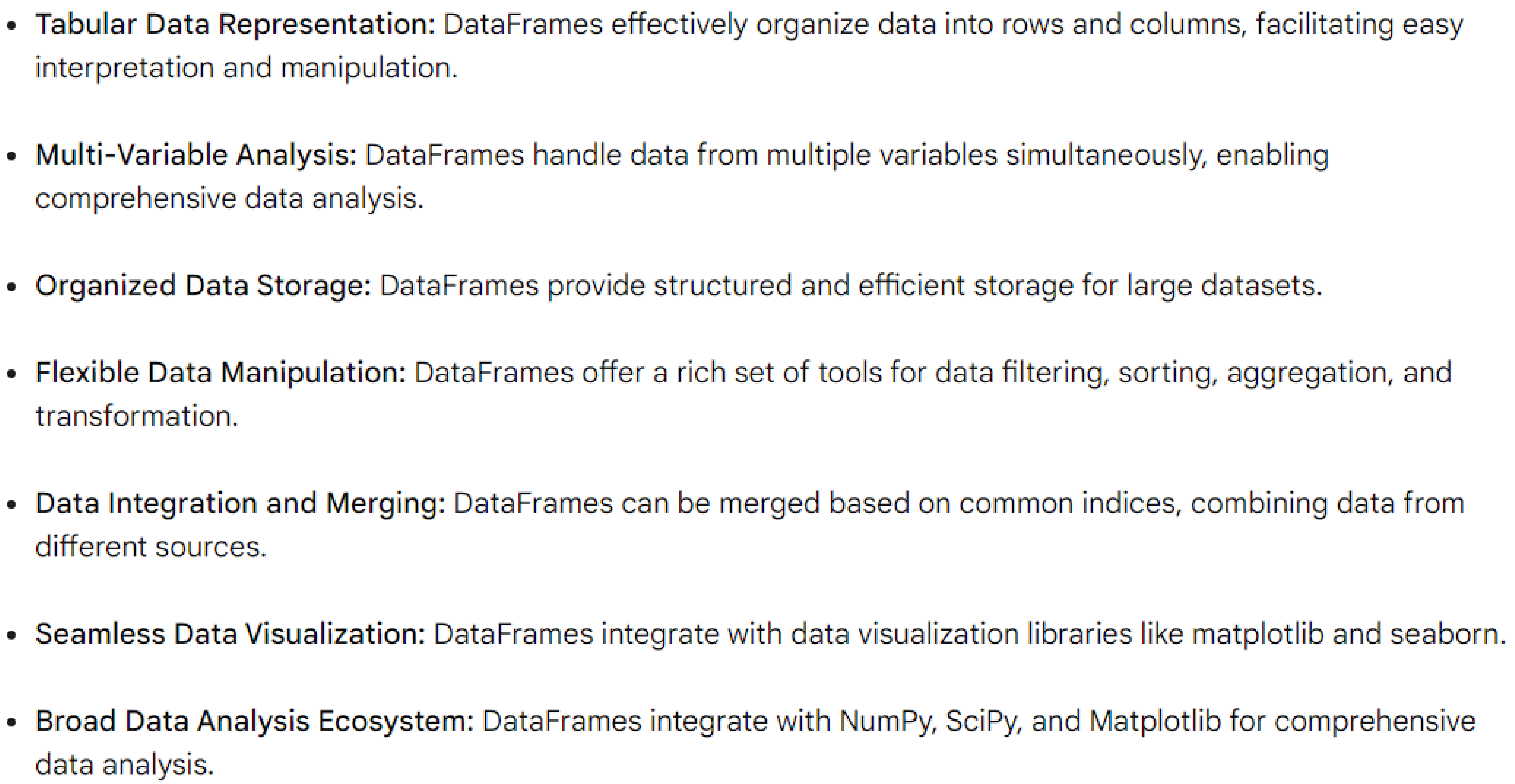
Data Structure in Pandas



Feature	Series	DataFrames
Dimensionality	One-dimensional	Two-dimensional
Data Type	Single data type per Series	Can store multiple data types in columns
Structure	Labeled array	Labeled tabular data
Usage	Representing time series data or cross-sectional data	Storing and analyzing complex datasets with multiple variables and observations

Pandas Basics

Advantages of Pandas Data Frame



- **Tabular Data Representation:** DataFrames effectively organize data into rows and columns, facilitating easy interpretation and manipulation.

- **Multi-Variable Analysis:** DataFrames handle data from multiple variables simultaneously, enabling comprehensive data analysis.

- **Organized Data Storage:** DataFrames provide structured and efficient storage for large datasets.

- **Flexible Data Manipulation:** DataFrames offer a rich set of tools for data filtering, sorting, aggregation, and transformation.

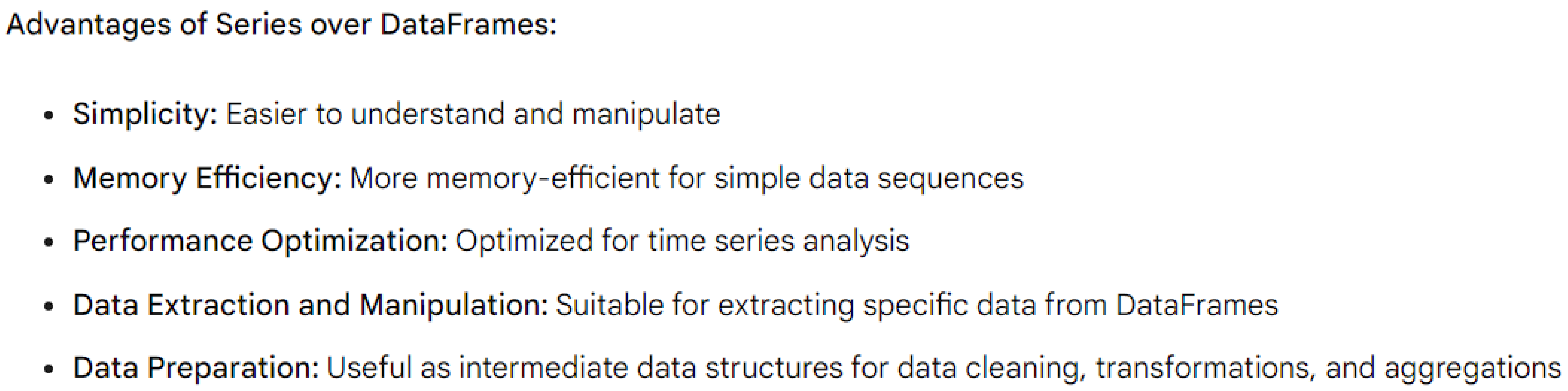
- **Data Integration and Merging:** DataFrames can be merged based on common indices, combining data from different sources.

- **Seamless Data Visualization:** DataFrames integrate with data visualization libraries like matplotlib and seaborn.

- **Broad Data Analysis Ecosystem:** DataFrames integrate with NumPy, SciPy, and Matplotlib for comprehensive data analysis.

Pandas Basics

Advantages of Series over Data Frame

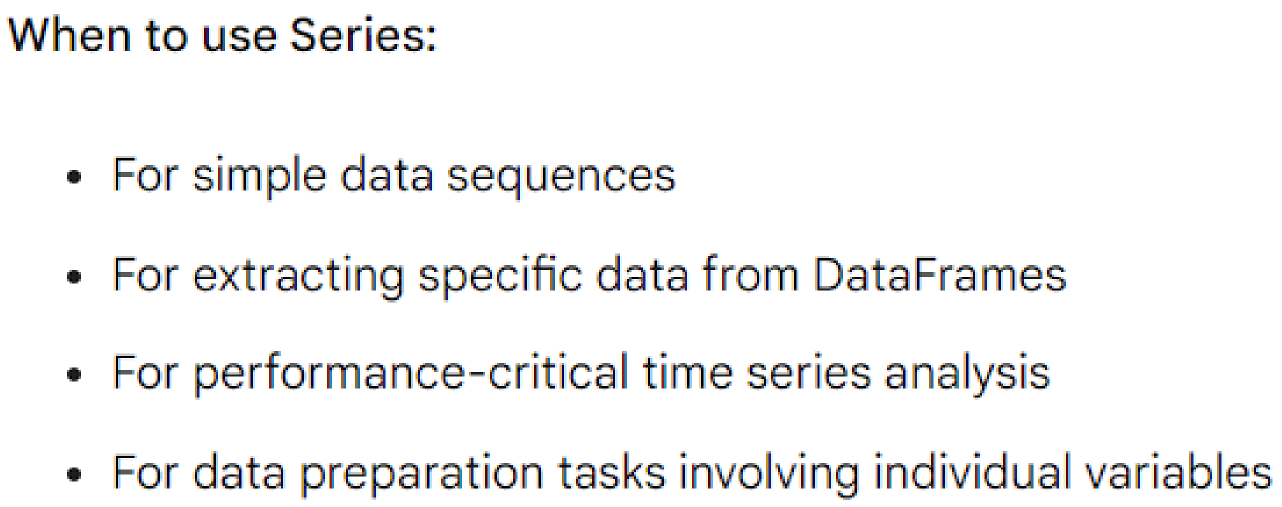


Advantages of Series over DataFrames:

- **Simplicity:** Easier to understand and manipulate
- **Memory Efficiency:** More memory-efficient for simple data sequences
- **Performance Optimization:** Optimized for time series analysis
- **Data Extraction and Manipulation:** Suitable for extracting specific data from DataFrames
- **Data Preparation:** Useful as intermediate data structures for data cleaning, transformations, and aggregations

Pandas Basics

When to use Series?

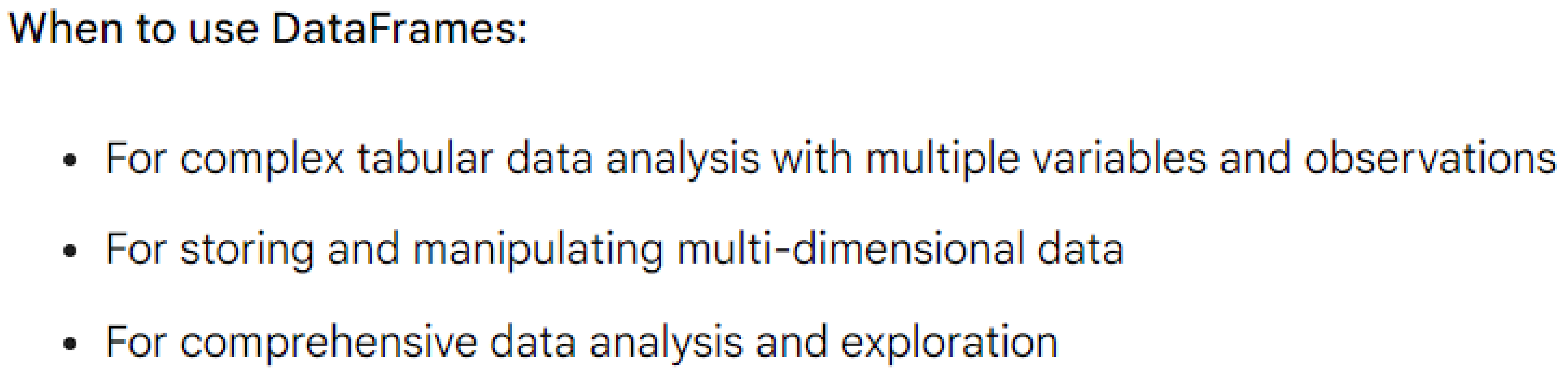


When to use Series:

- For simple data sequences
- For extracting specific data from DataFrames
- For performance-critical time series analysis
- For data preparation tasks involving individual variables

Pandas Basics

When to use DataFrame?



When to use DataFrames:

- For complex tabular data analysis with multiple variables and observations
- For storing and manipulating multi-dimensional data
- For comprehensive data analysis and exploration

www.aiquest.org

PANDAS

শেখার কোন শেষ নাই!

NO Ads!

5 Mins to Pandas

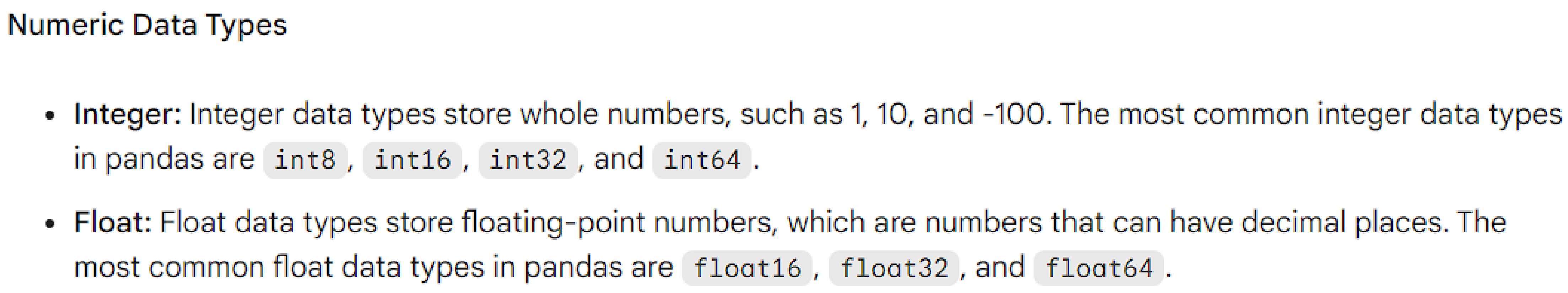
Instructor:
Rashedul Alam Shakil
Founder of aiQuest Intelligence
M.Sc. in Data Science at FAU Germany

5 Mins to Pandas

Pandas Basics

Datatypes in Pandas

Pandas is a powerful Python library for data analysis. One of its key features is its ability to handle different data types effectively. Pandas data types can be broadly categorized into three main types:

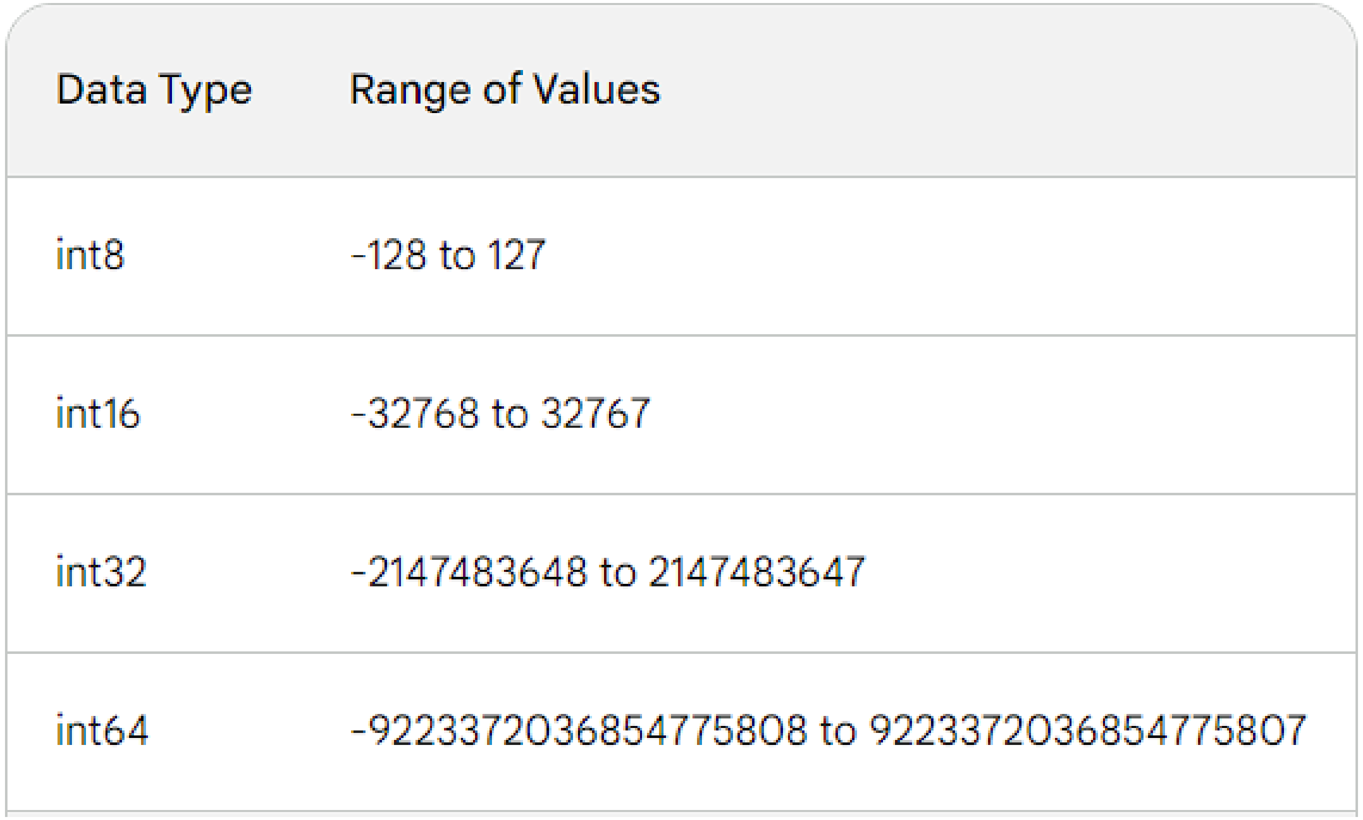


Numeric Data Types

- **Integer:** Integer data types store whole numbers, such as 1, 10, and -100. The most common integer data types in pandas are `int8`, `int16`, `int32`, and `int64`.
- **Float:** Float data types store floating-point numbers, which are numbers that can have decimal places. The most common float data types in pandas are `float16`, `float32`, and `float64`.

Pandas Basics

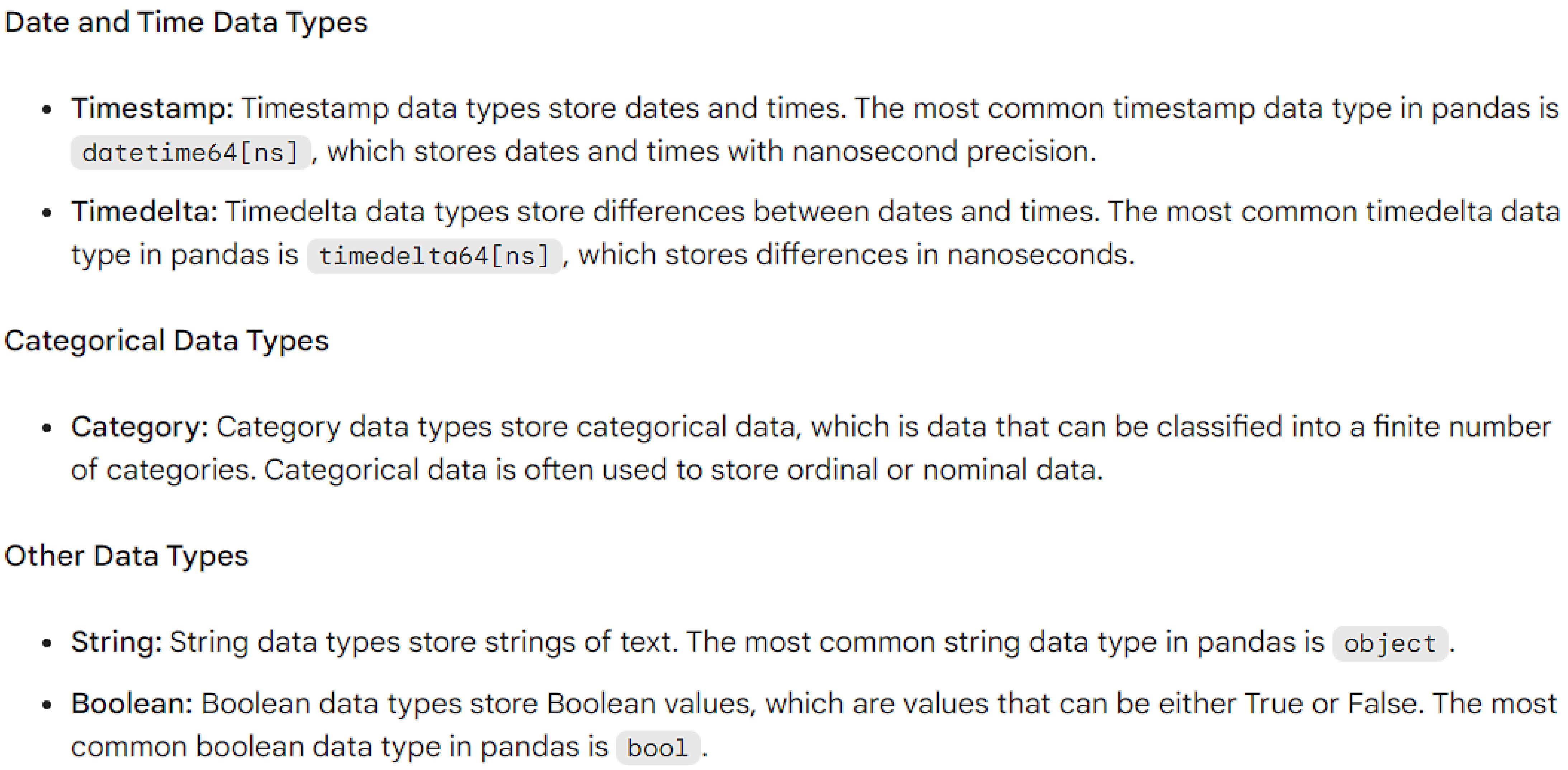
Datatypes in Pandas



Data Type	Range of Values
int8	-128 to 127
int16	-32768 to 32767
int32	-2147483648 to 2147483647
int64	-9223372036854775808 to 9223372036854775807

Pandas Basics

Datatypes in Pandas



Date and Time Data Types

- **Timestamp:** Timestamp data types store dates and times. The most common timestamp data type in pandas is `datetime64[ns]`, which stores dates and times with nanosecond precision.
- **Timedelta:** Timedelta data types store differences between dates and times. The most common timedelta data type in pandas is `timedelta64[ns]`, which stores differences in nanoseconds.

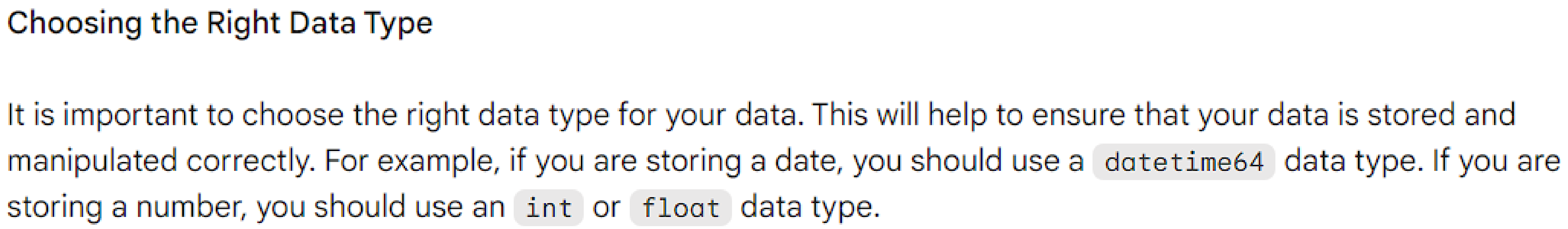
Categorical Data Types

- **Category:** Category data types store categorical data, which is data that can be classified into a finite number of categories. Categorical data is often used to store ordinal or nominal data.

Other Data Types

- **String:** String data types store strings of text. The most common string data type in pandas is `object`.
- **Boolean:** Boolean data types store Boolean values, which are values that can be either True or False. The most common boolean data type in pandas is `bool`.

How to Choose the Right Datatype?



Choosing the Right Data Type

It is important to choose the right data type for your data. This will help to ensure that your data is stored and manipulated correctly. For example, if you are storing a date, you should use a `datetime64` data type. If you are storing a number, you should use an `int` or `float` data type.