BHARATHIDASAN UNIVERSITY

Tiruchirappalli- 620024

Tamil Nadu, India

Programme: MBA (Financial Management)

Course Title : R and Python for Finance (NSE)

Course Code : FMEC2/24

Unit III : Financial Time Series

Dr. M. KAMAL
Visiting Faculty
Asst. Professor of Computer Applications
Jamal Mohamed College
Tiruchirappalli-20

Pandas: A Powerful Tool for Data Analysis

Pandas is a Python library that provides high-performance, easy-to-use data structures and data analysis tools. It is built on top of NumPy and is widely used in data science, machine learning, and financial analysis.

DataFrame: The Core Data Structure

Rows and Columns

Each row represents an observation, and each column represents a feature or variable.

Heterogeneous Data

DataFrames can store data of different types, such as integers, floats, strings, and more.

Indexing

Rows and columns can be accessed using labels or integer-based indexing.

Essential DataFrame Operations

1 **Inspecting Data**

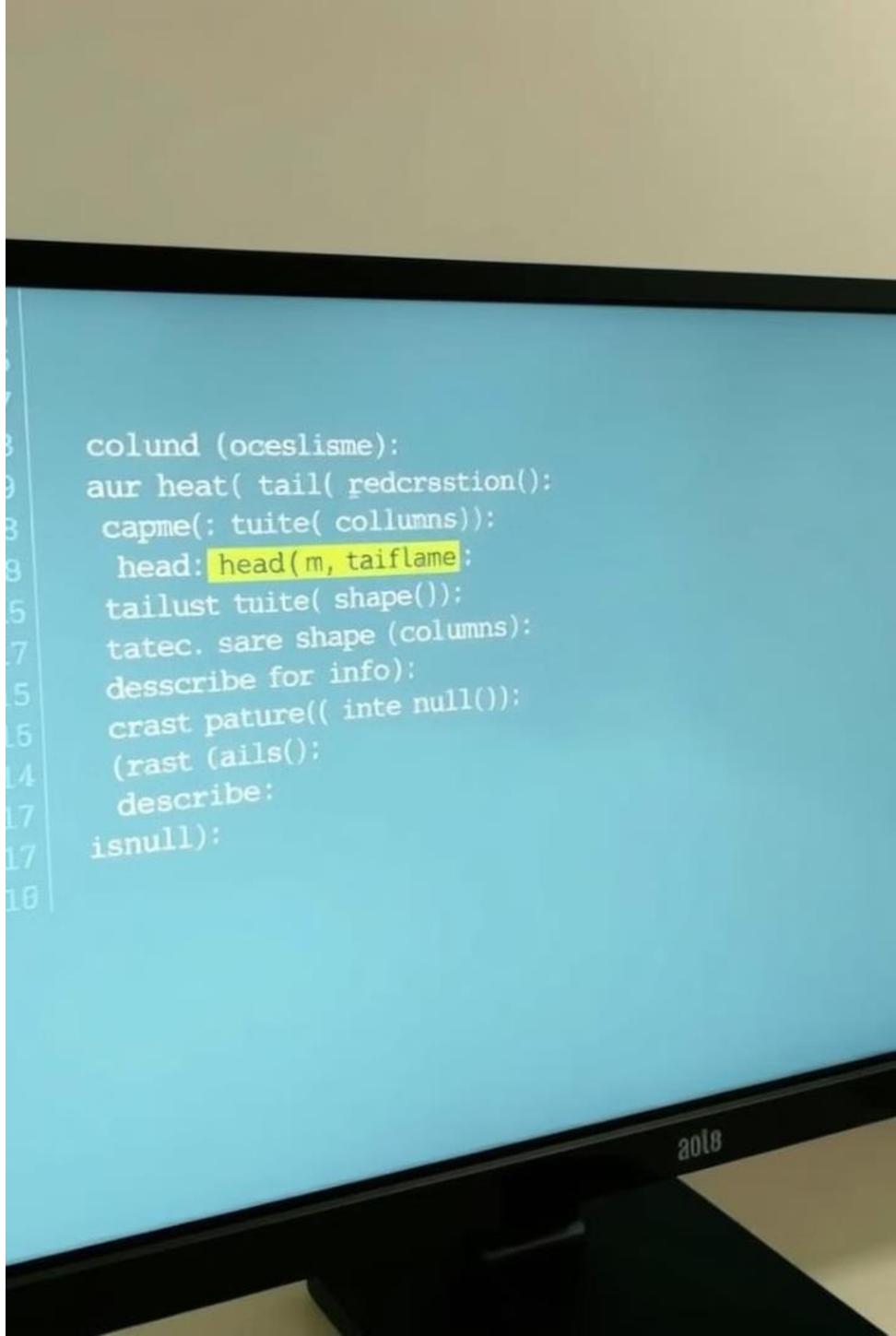
Use `head()`, `tail()`, and `shape()` to view the first few rows, last few rows, and dimensions of the DataFrame.

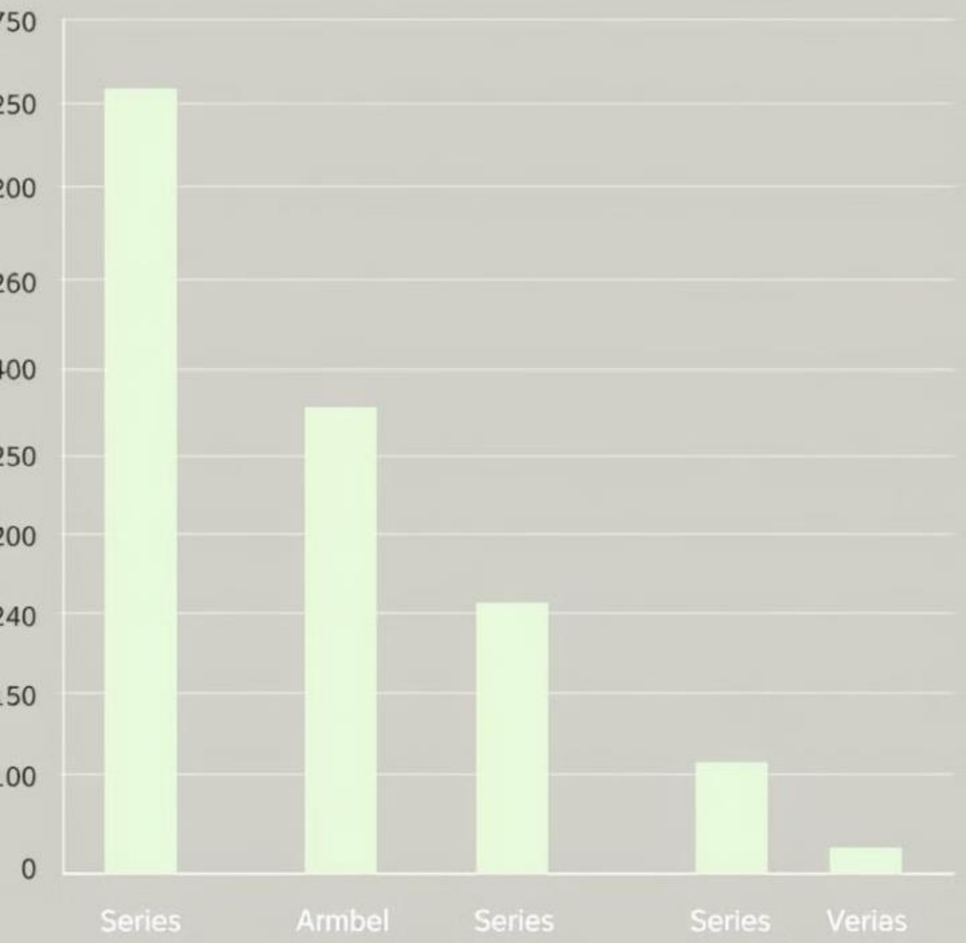
2 **Data Summary**

Use `info()` and `describe()` to get a summary of the DataFrame's structure and statistical properties.

3 **Handling Missing Values**

Use `isnull().sum()` to count missing values in each column.





Series: One-Dimensional Data

The Series class is a one-dimensional, labeled array that can hold any data type. It is similar to a column in a DataFrame.

Group By Operations: Aggregating Data

Split

Group the data based on a column or criteria.

Apply

Apply an aggregation function, such as `sum()`, `mean()`, or `max()`, to each group.

Combine

Produce the result as a new DataFrame or Series.

Regression Analysis: Modeling Relationships

Linear Regression

For linear relationships between variables.

Multiple Linear Regression

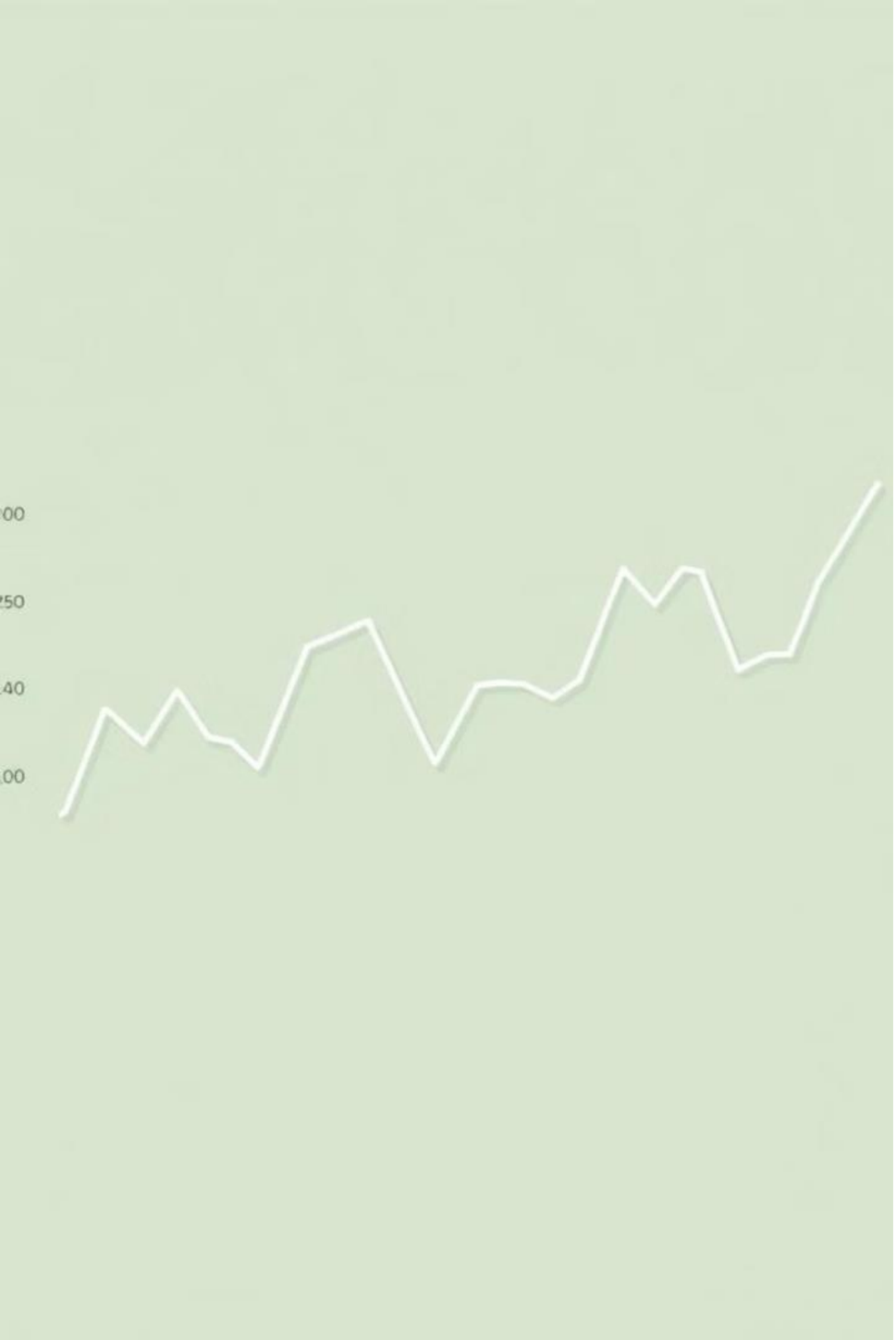
With multiple independent variables.

Polynomial Regression

For non-linear relationships.

Logistic Regression

For binary classification problems.



High-Frequency Data: Challenges and Solutions

Large Volume

Efficient processing is essential for handling massive datasets.

Irregular Time Intervals

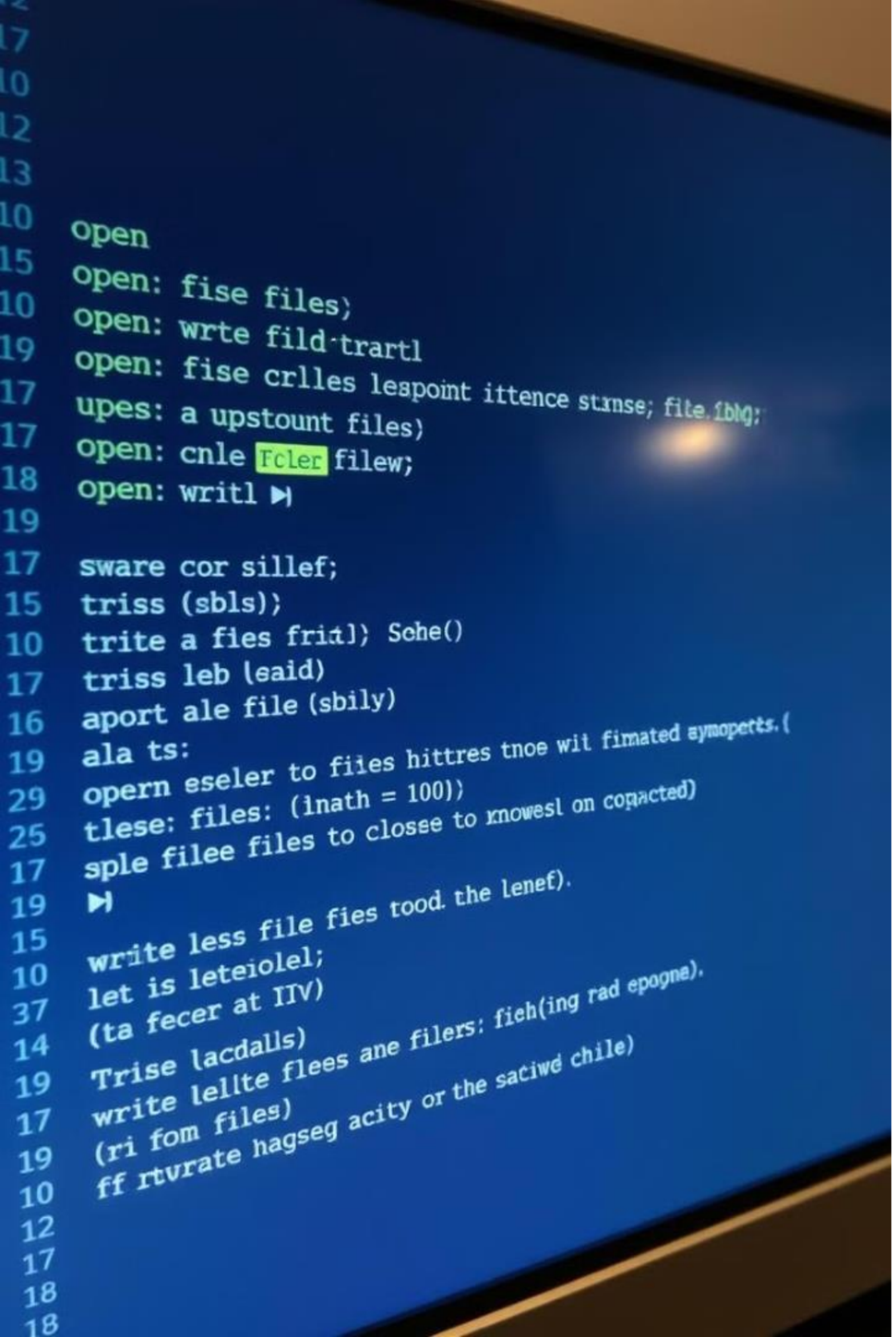
Data often has irregular timestamps, requiring careful handling.

Noise

High-frequency data often contains market microstructure noise.

Synchronization

Combining data from different sources or assets can be challenging.



Basic I/O with Python: Reading and Writing Data

1 **File Input/Output**

Use the `open()` function with different modes for reading, writing, and appending data to files.

2 **Reading from a File**

Use the `read()` method to read the entire contents of a file or the `readline()` method to read one line at a time.

3 **Writing to a File**

Use the `write()` method to write data to a file.

Connecting to SQL Databases

1

Import Module

Import the necessary module, such as `mysql.connector` or `PyMySQL`.

2

Establish Connection

Create a connection object to the database.

3

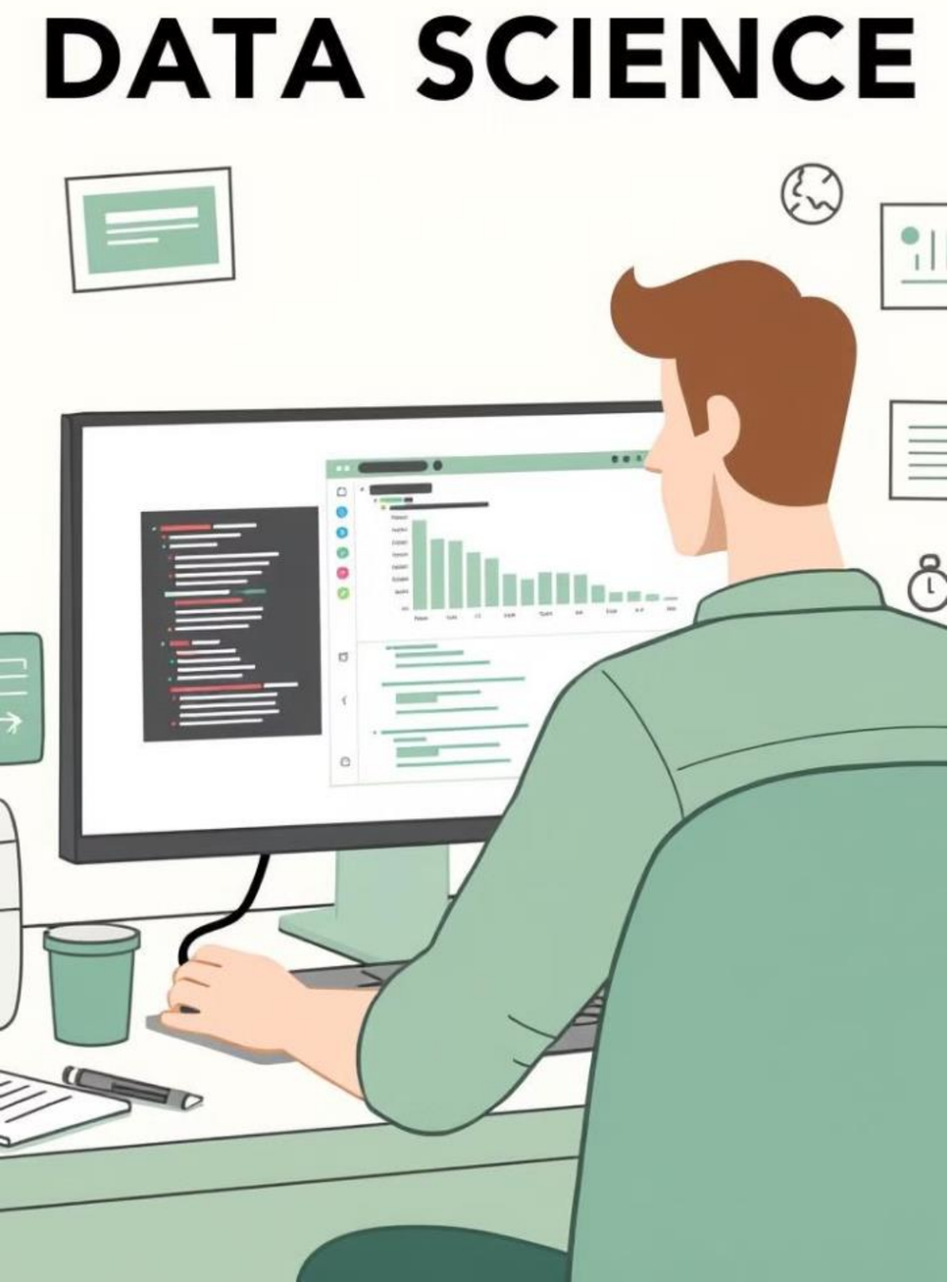
Create Cursor

Create a cursor object to execute SQL queries.

4

Execute Queries

Use the cursor object to execute SQL queries, such as `CREATE TABLE`, `INSERT INTO`, or `SELECT`.



Key Takeaways and Next Steps

Pandas is a powerful library for data analysis in Python. It provides a wide range of features for data manipulation, aggregation, and visualization. To further explore Pandas, consider learning about advanced data manipulation techniques, time series analysis, and data visualization libraries.