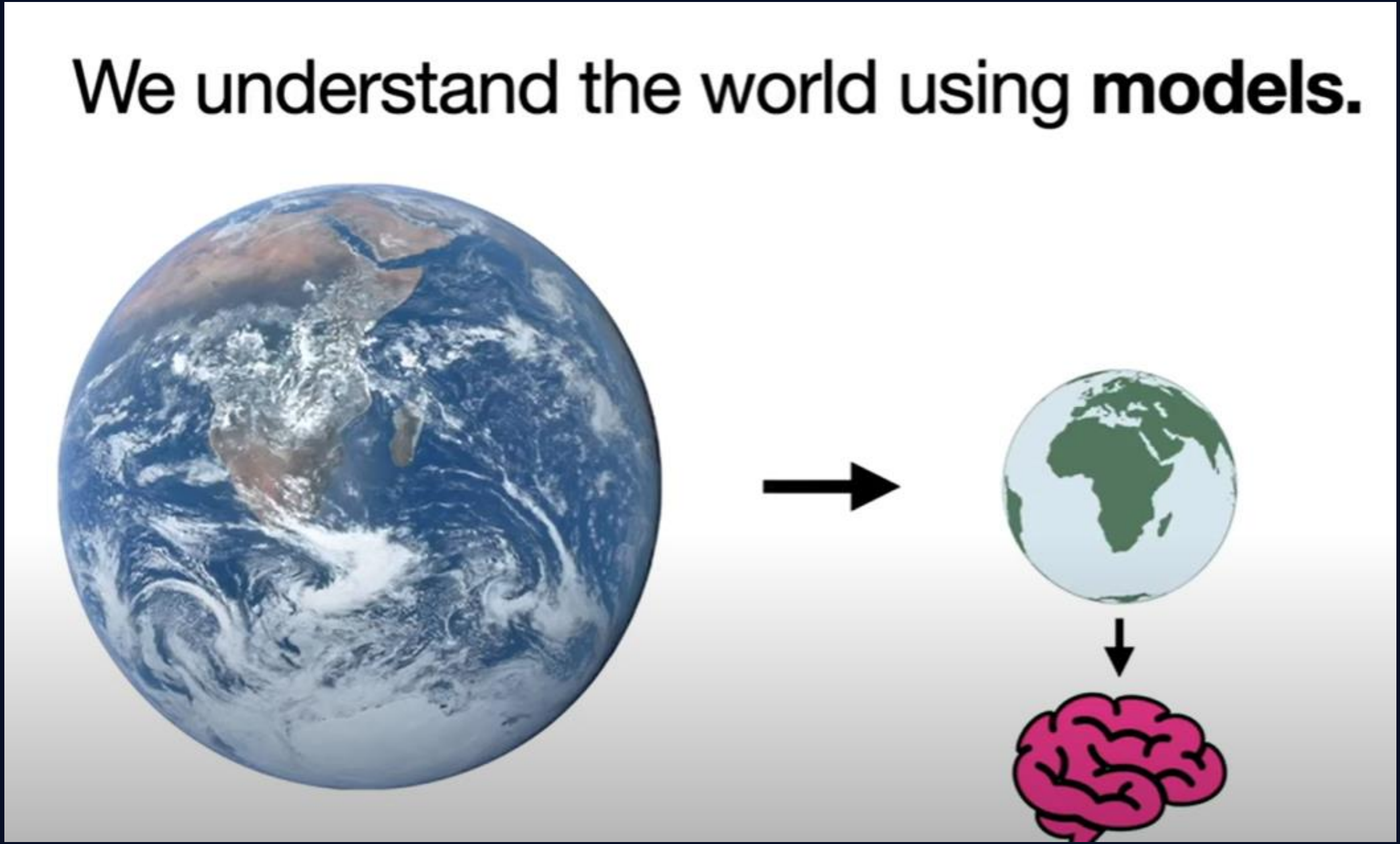
DATA ANALYSIS USING PANDAS

Dr Sivabalan,

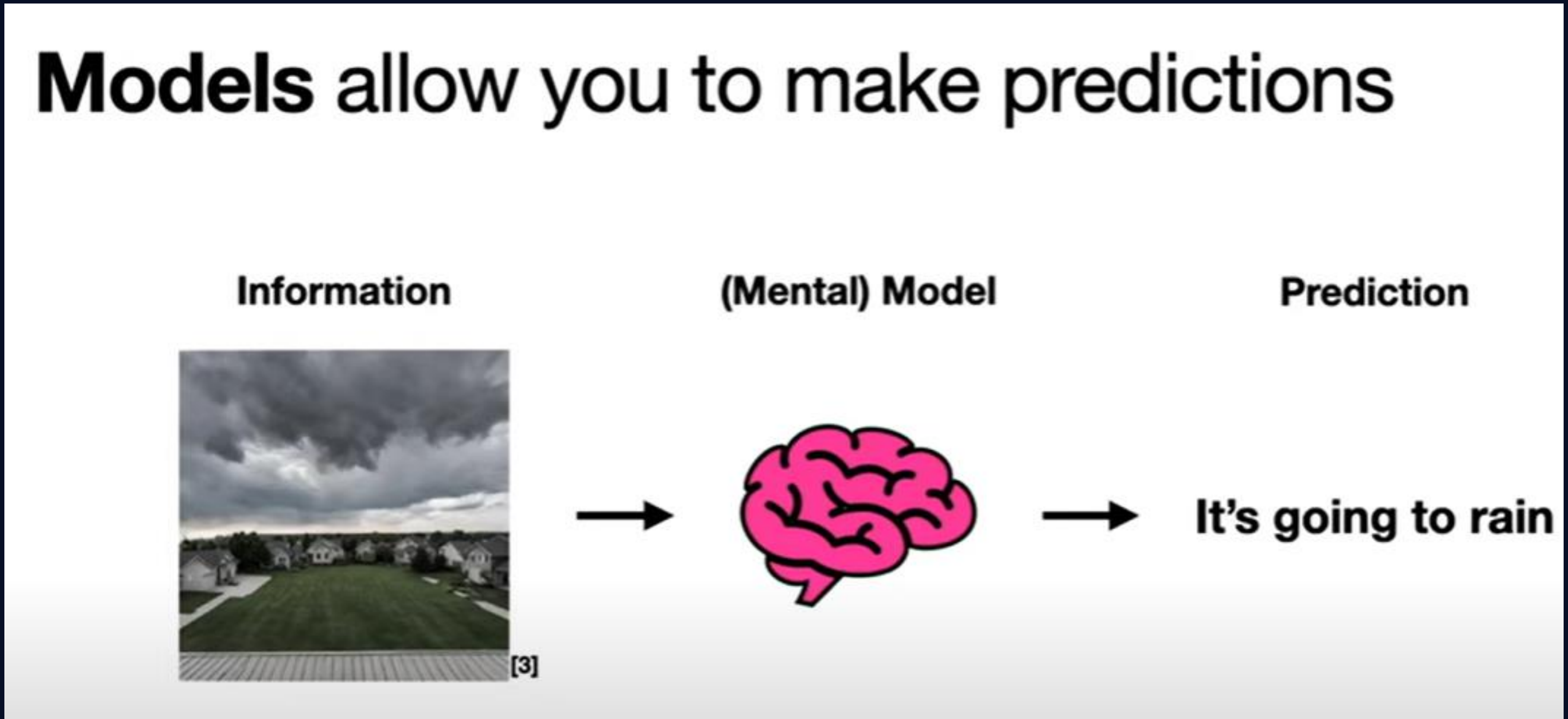
Technical Training Advisor

Sivabalan.n@nttdata.com



We understand the world using **models.**

sa



Models allow you to make predictions

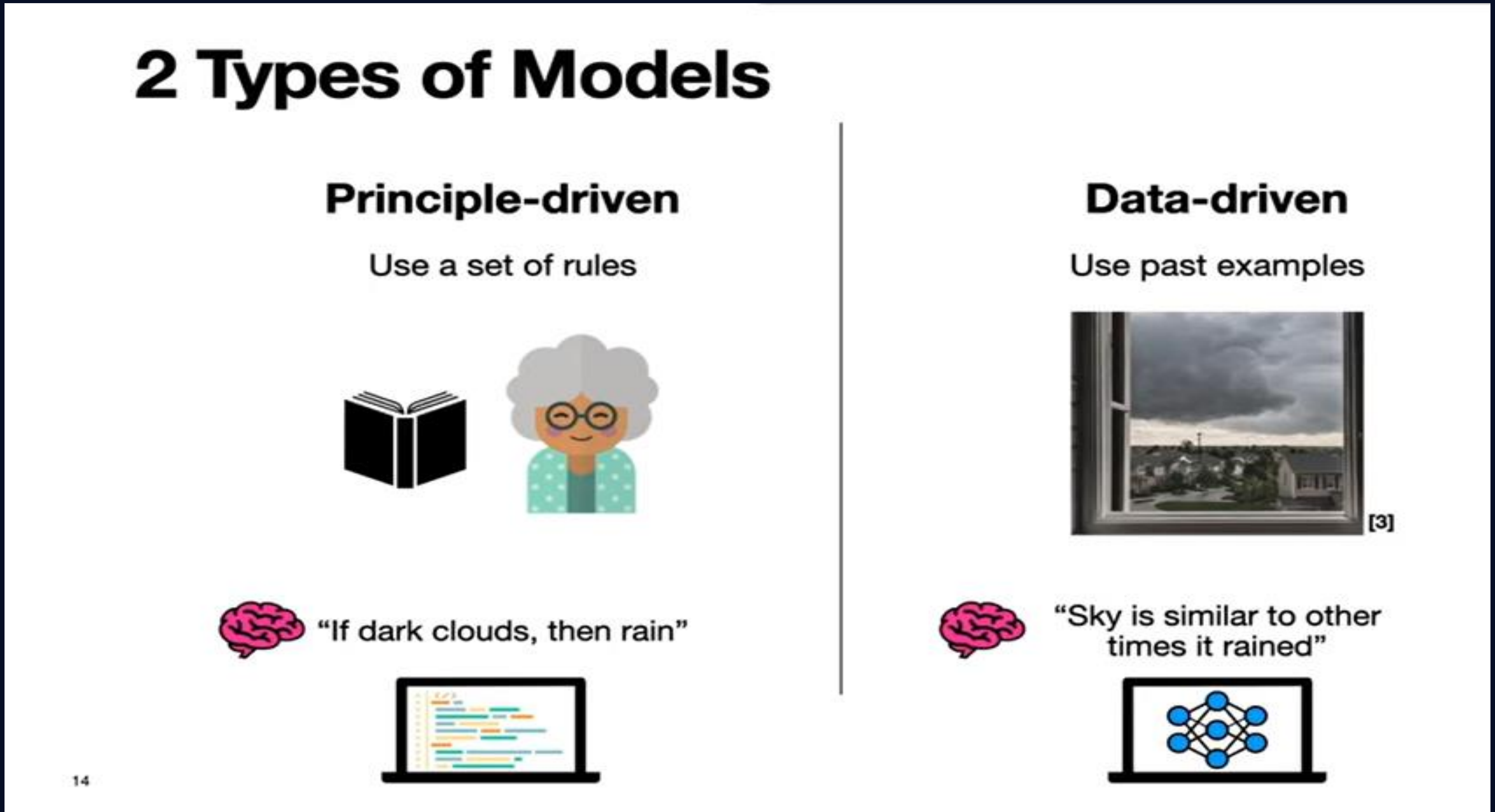
Information → (Mental) Model → Prediction

[3]

It's going to rain

Where do models come from?

sa



2 Types of Models

Principle-driven

Use a set of rules

"If dark clouds, then rain"

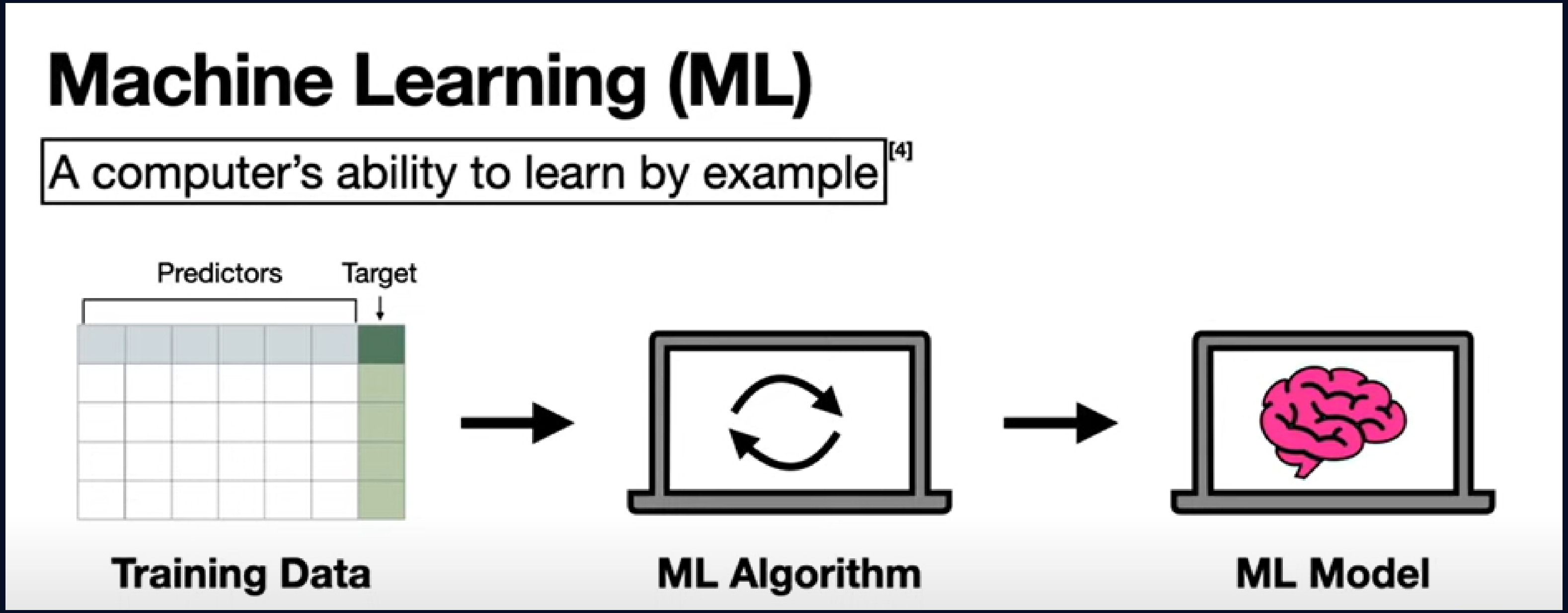
Data-driven

Use past examples

[3]

"Sky is similar to other times it rained"

sa



Machine Learning (ML)

A computer's ability to learn by example[4]

Predictors

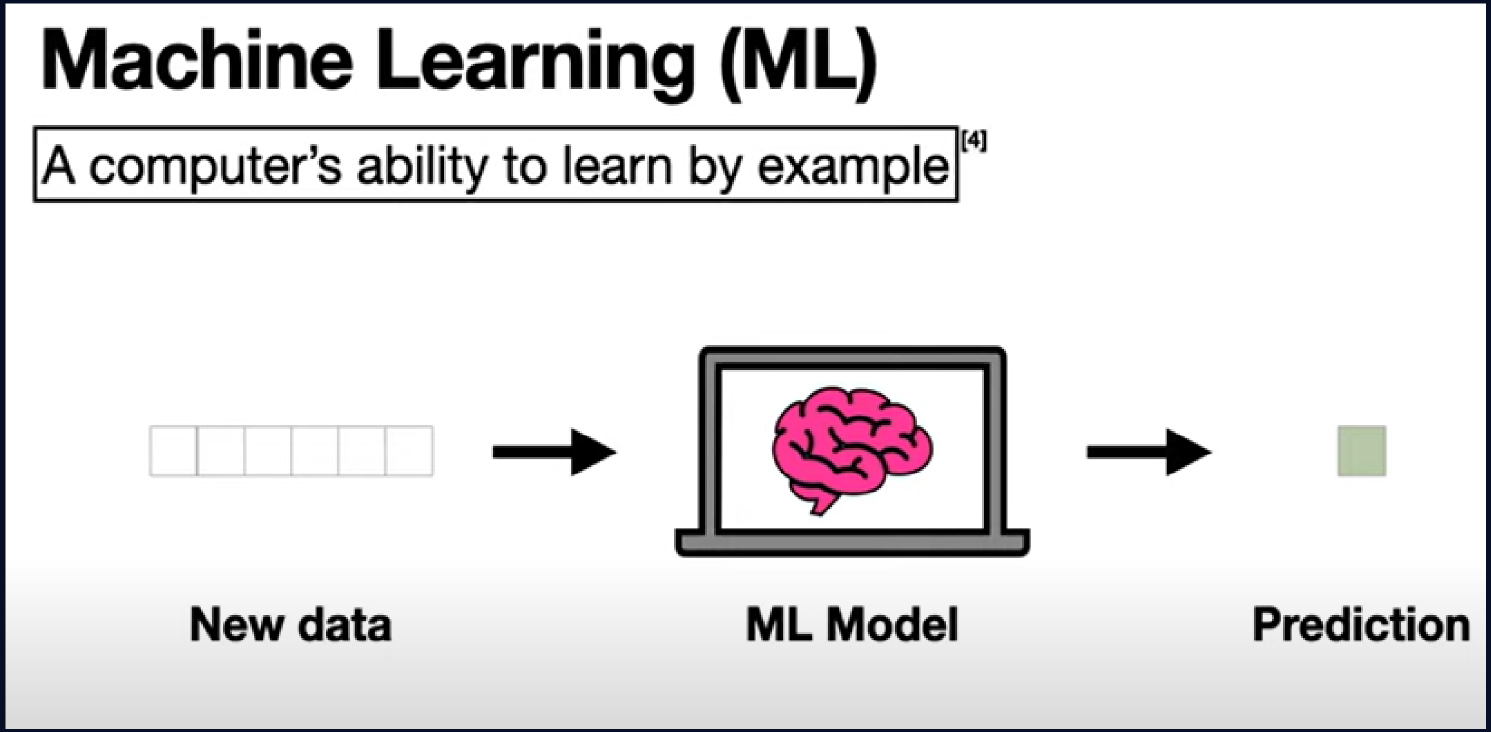
Target

Training Data

ML Algorithm

ML Model

sa



Machine Learning (ML)

A computer's ability to learn by example[4]

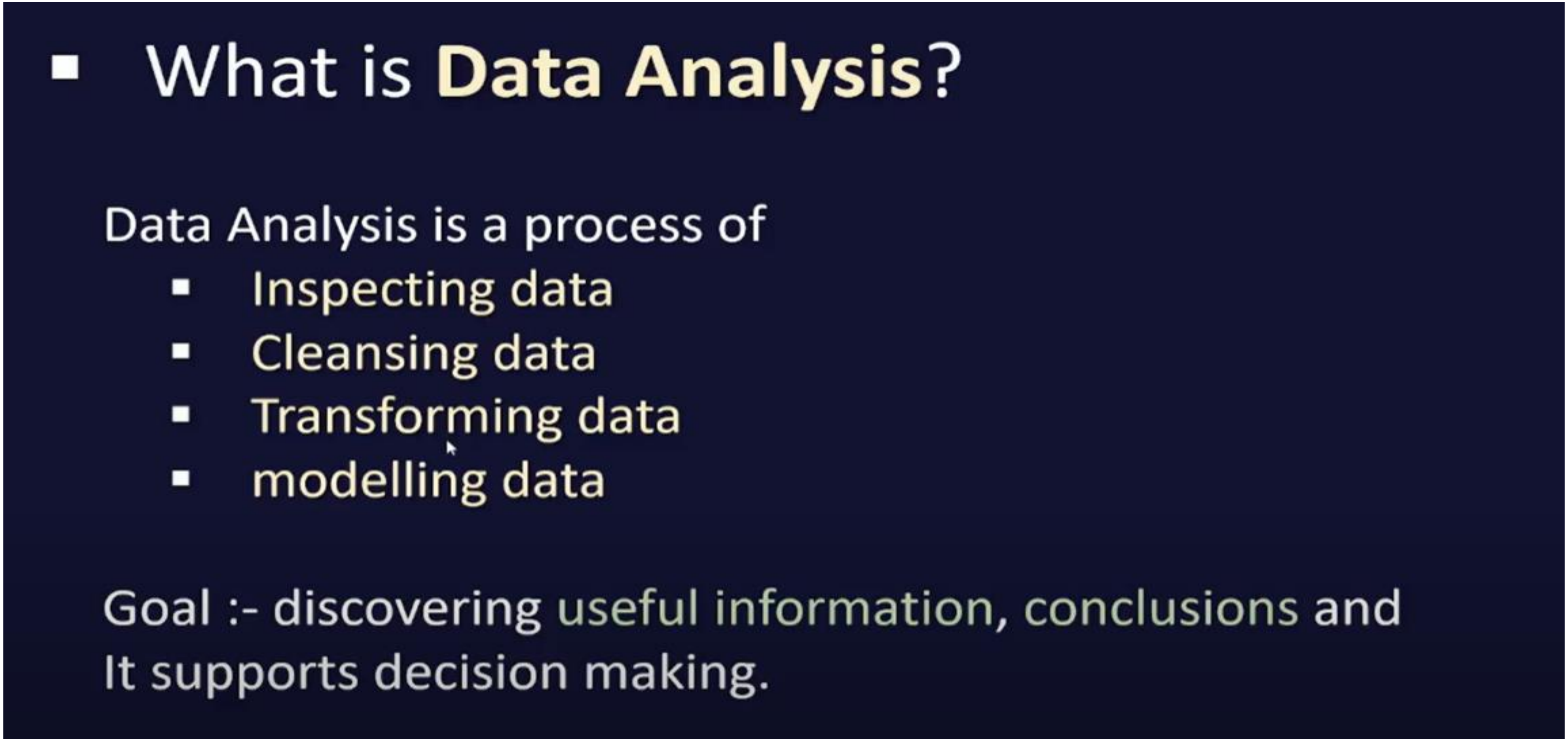
New data

ML Model

Prediction

sa

Data Analysis



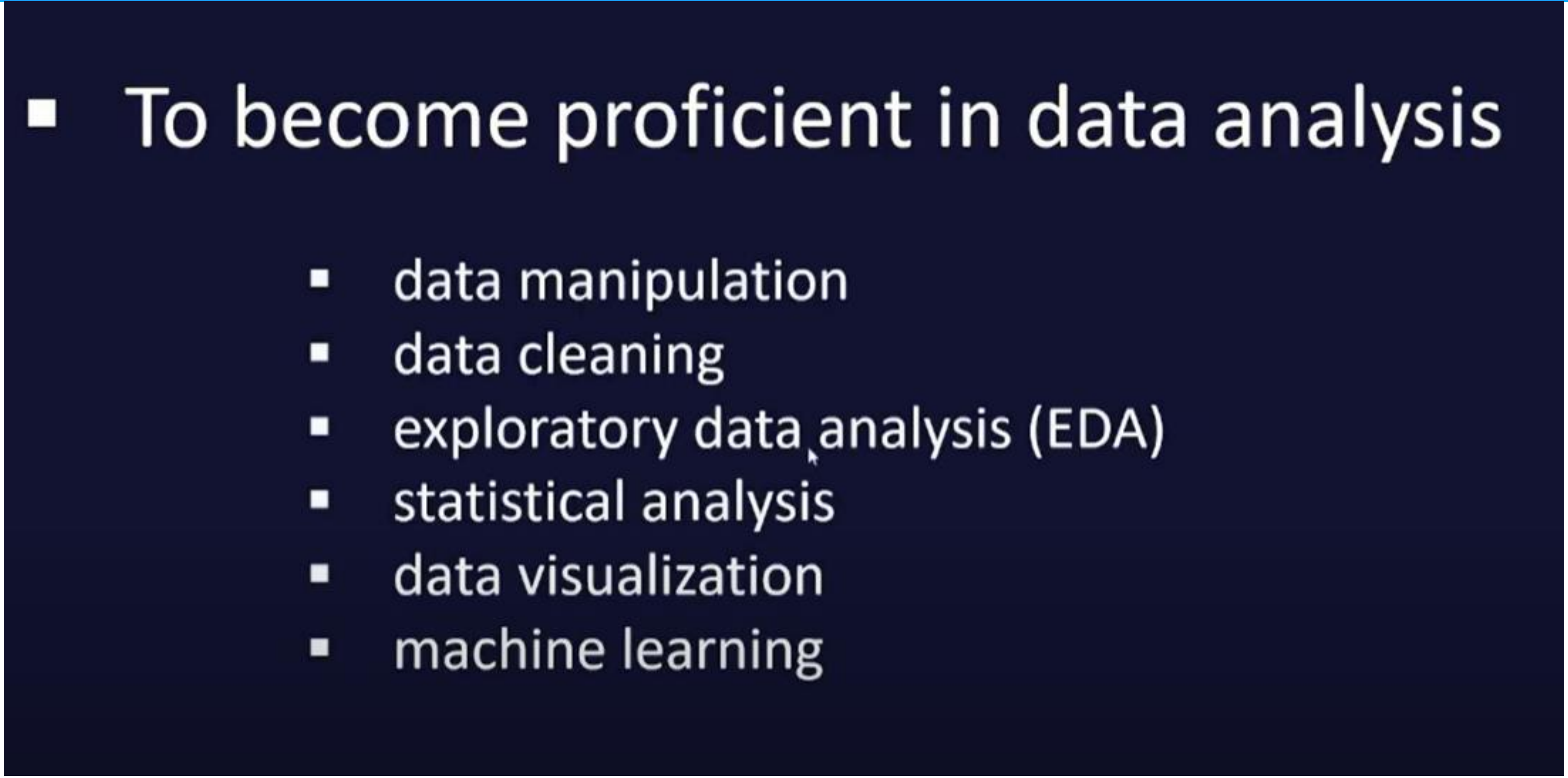
- What is **Data Analysis**?

Data Analysis is a process of

- Inspecting data
- Cleansing data
- Transforming data
- modelling data

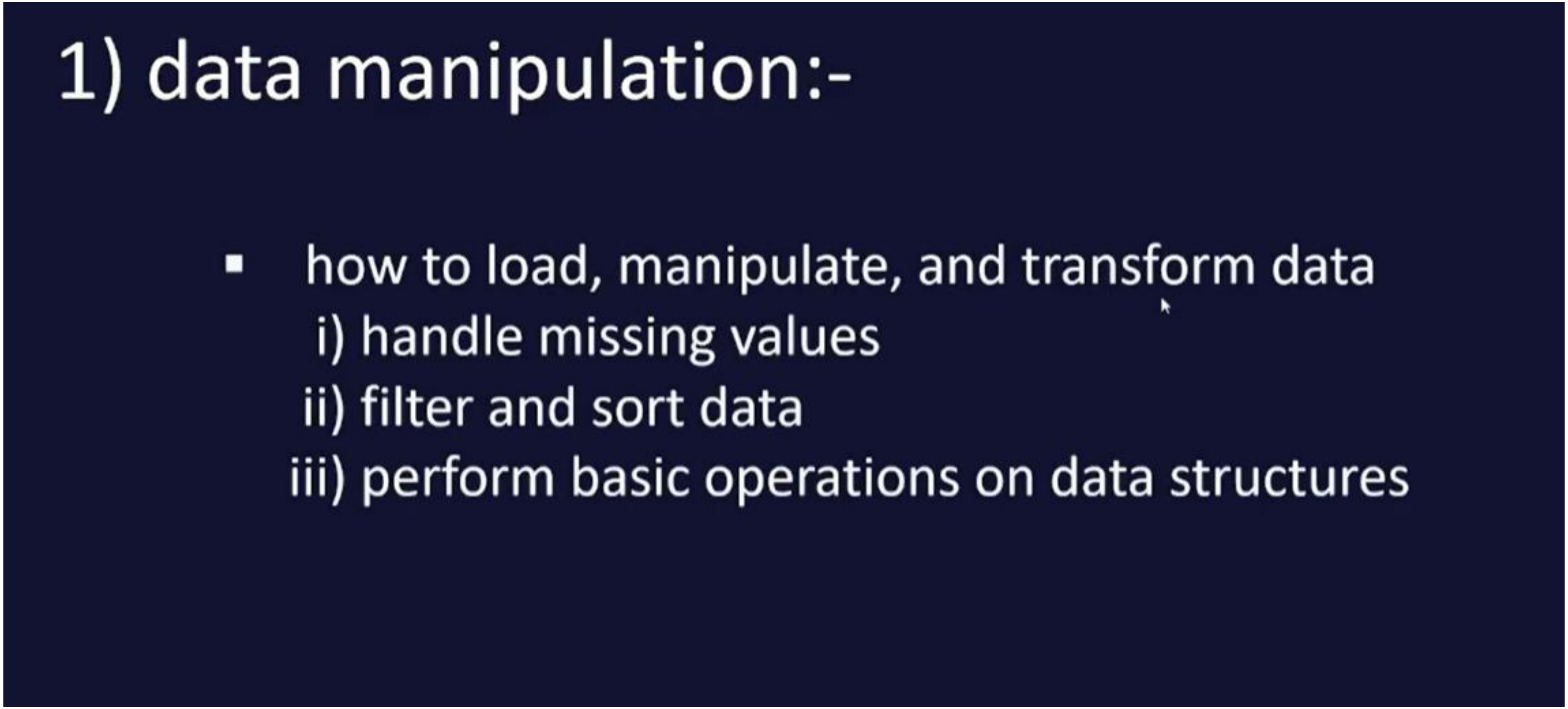
Goal :- discovering useful information, conclusions and It supports decision making.

Data Analysis



- To become proficient in data analysis
  - data manipulation
  - data cleaning
  - exploratory data analysis (EDA)
  - statistical analysis
  - data visualization
  - machine learning

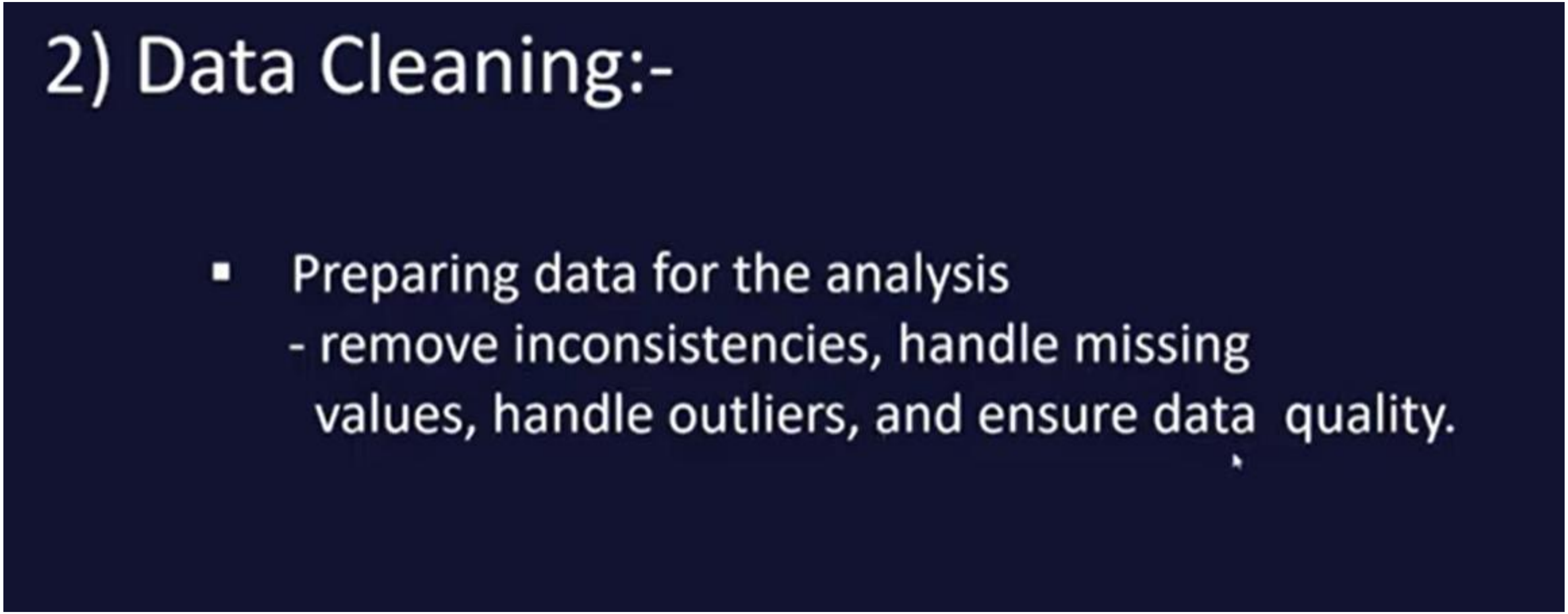
Data Analysis



1) data manipulation:-

- how to load, manipulate, and transform data
  i) handle missing values
  ii) filter and sort data
  iii) perform basic operations on data structures

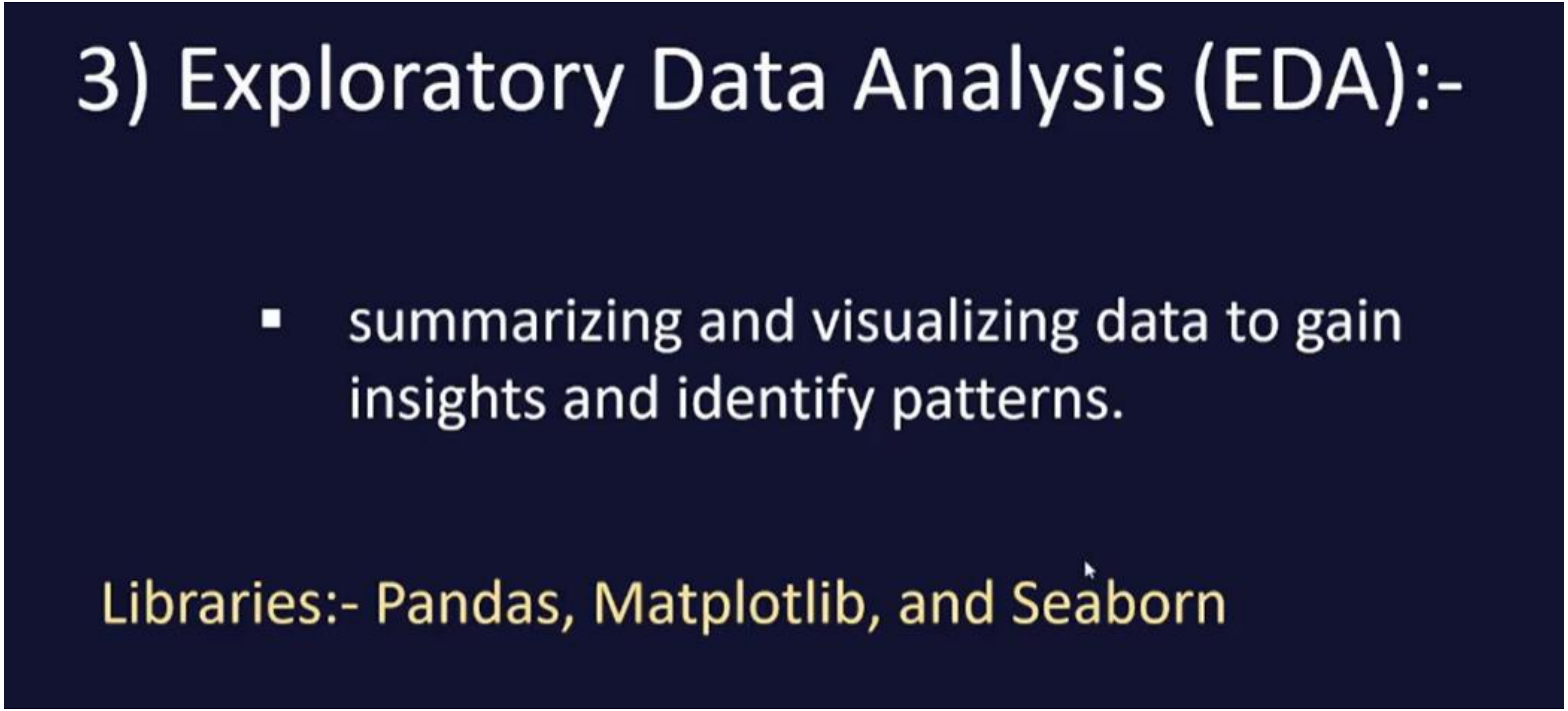
Data Analysis



2) Data Cleaning:-

- Preparing data for the analysis
  - remove inconsistencies, handle missing values, handle outliers, and ensure data quality.

Data Analysis

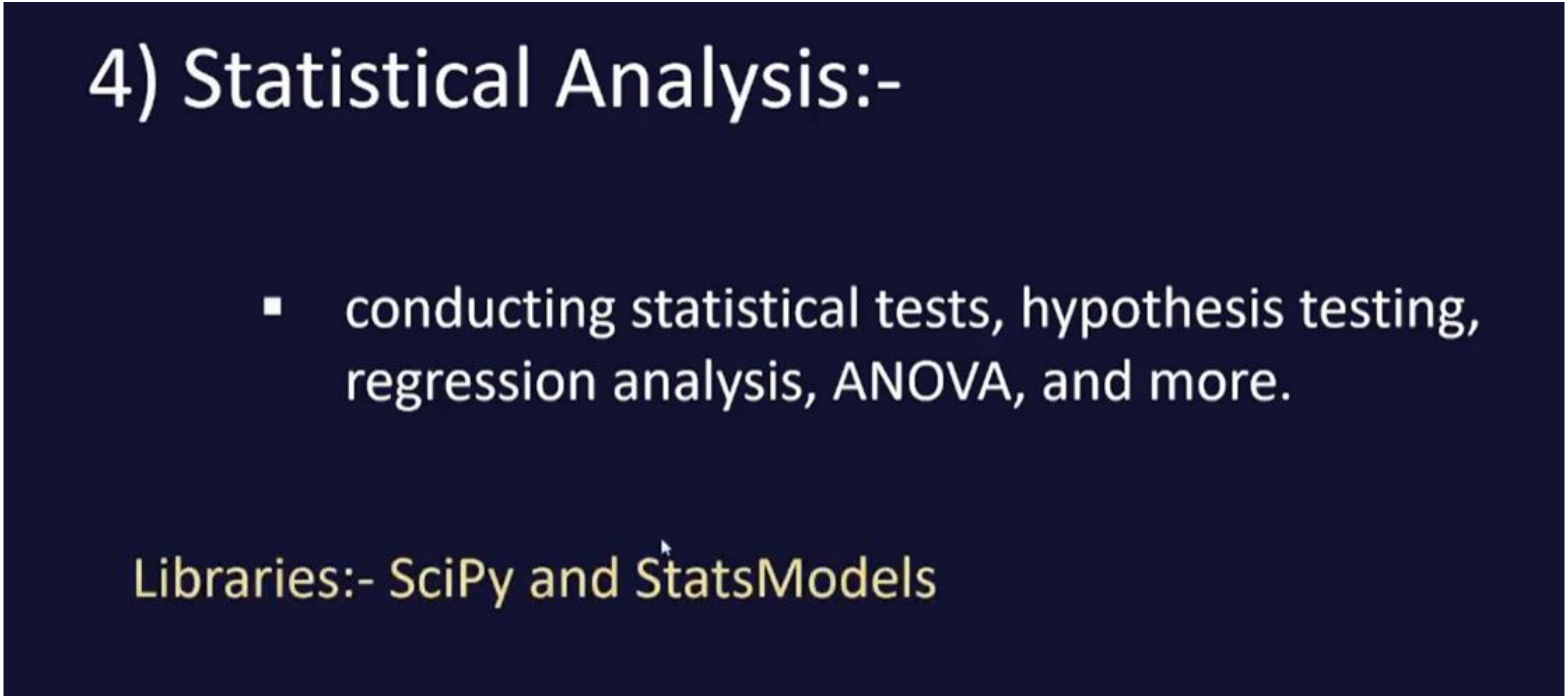


3) Exploratory Data Analysis (EDA):-

- summarizing and visualizing data to gain insights and identify patterns.

Libraries:- Pandas, Matplotlib, and Seaborn

Data Analysis

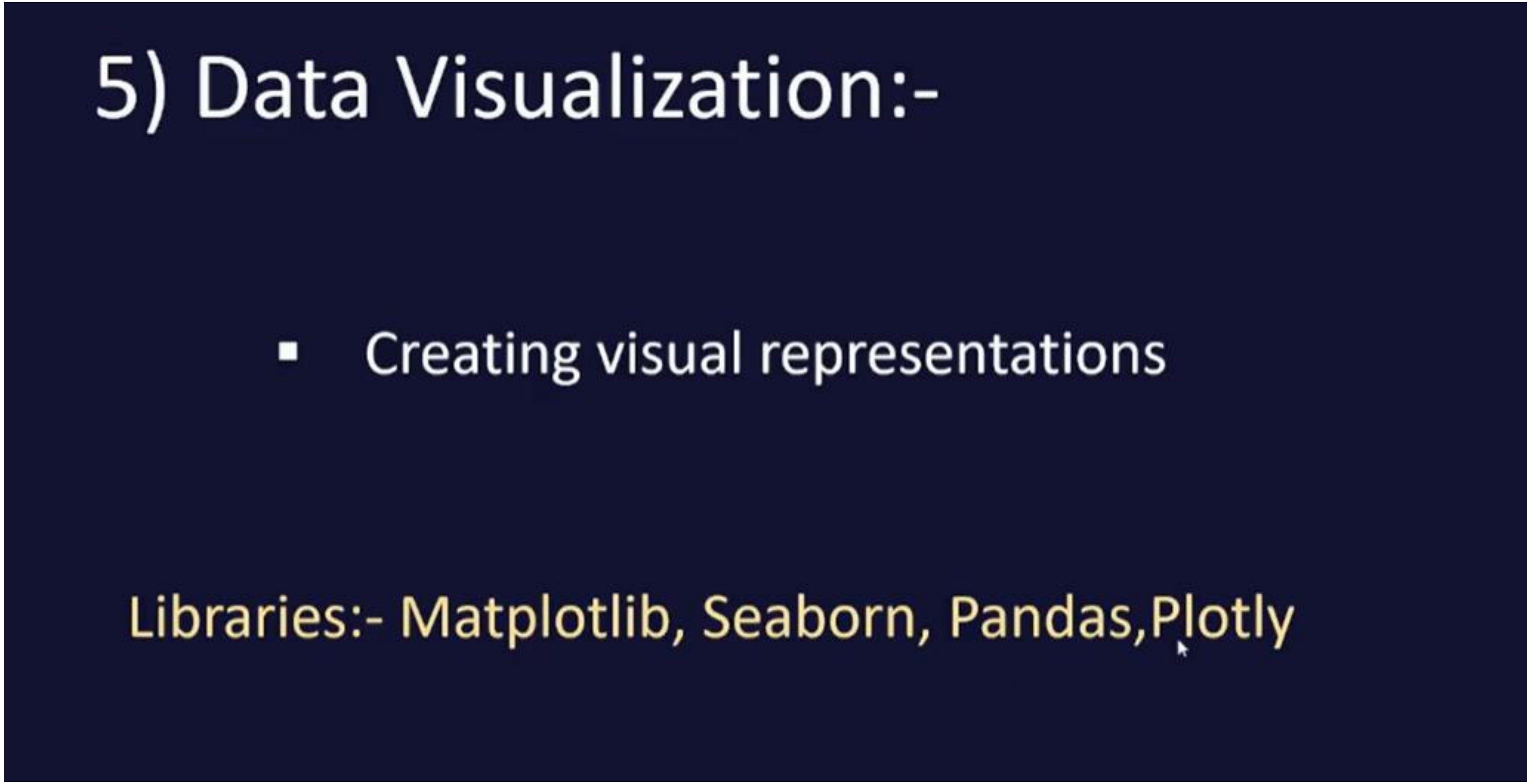


4) Statistical Analysis:-

- conducting statistical tests, hypothesis testing, regression analysis, ANOVA, and more.

Libraries:- SciPy and StatsModels

Data Analysis

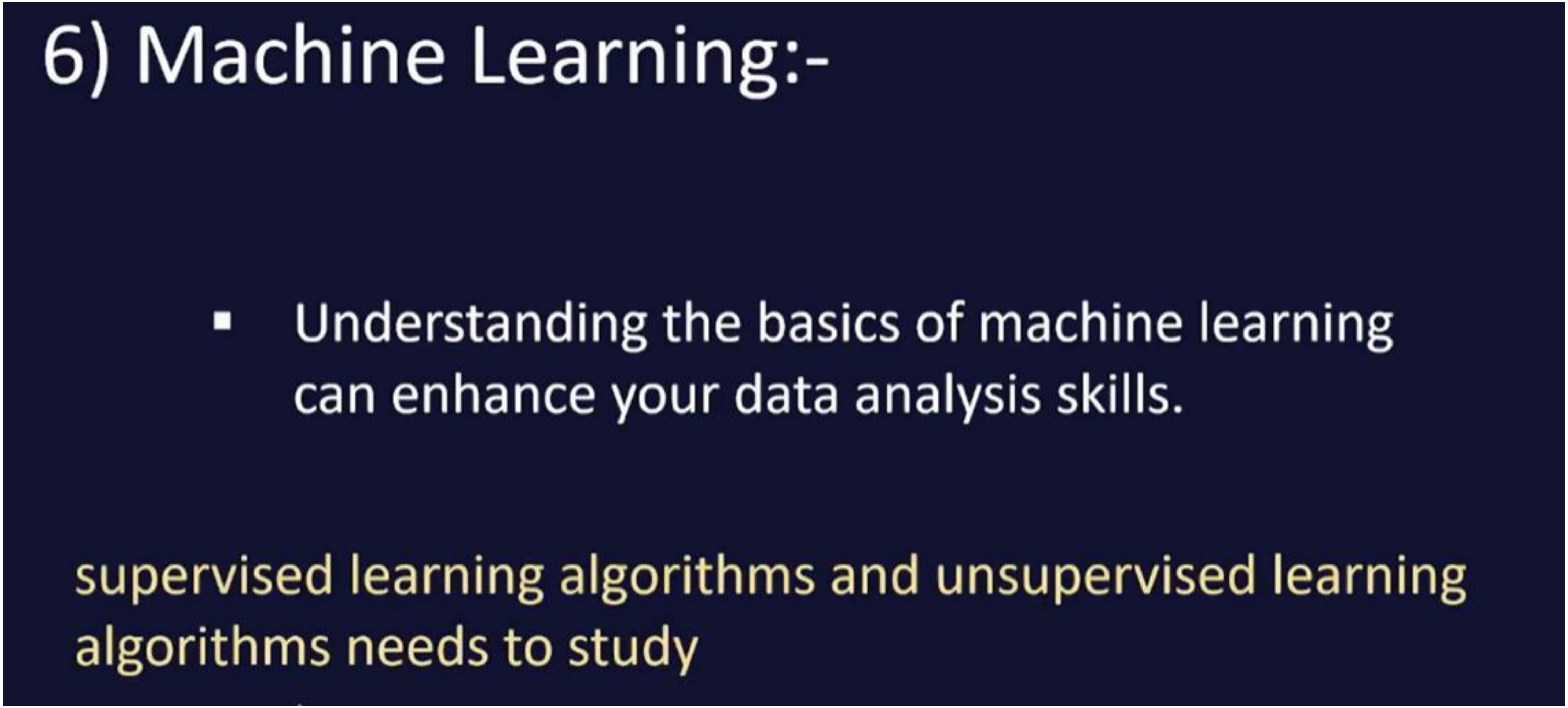


5) Data Visualization:-

- Creating visual representations

Libraries:- Matplotlib, Seaborn, Pandas,Plotly

Data Analysis

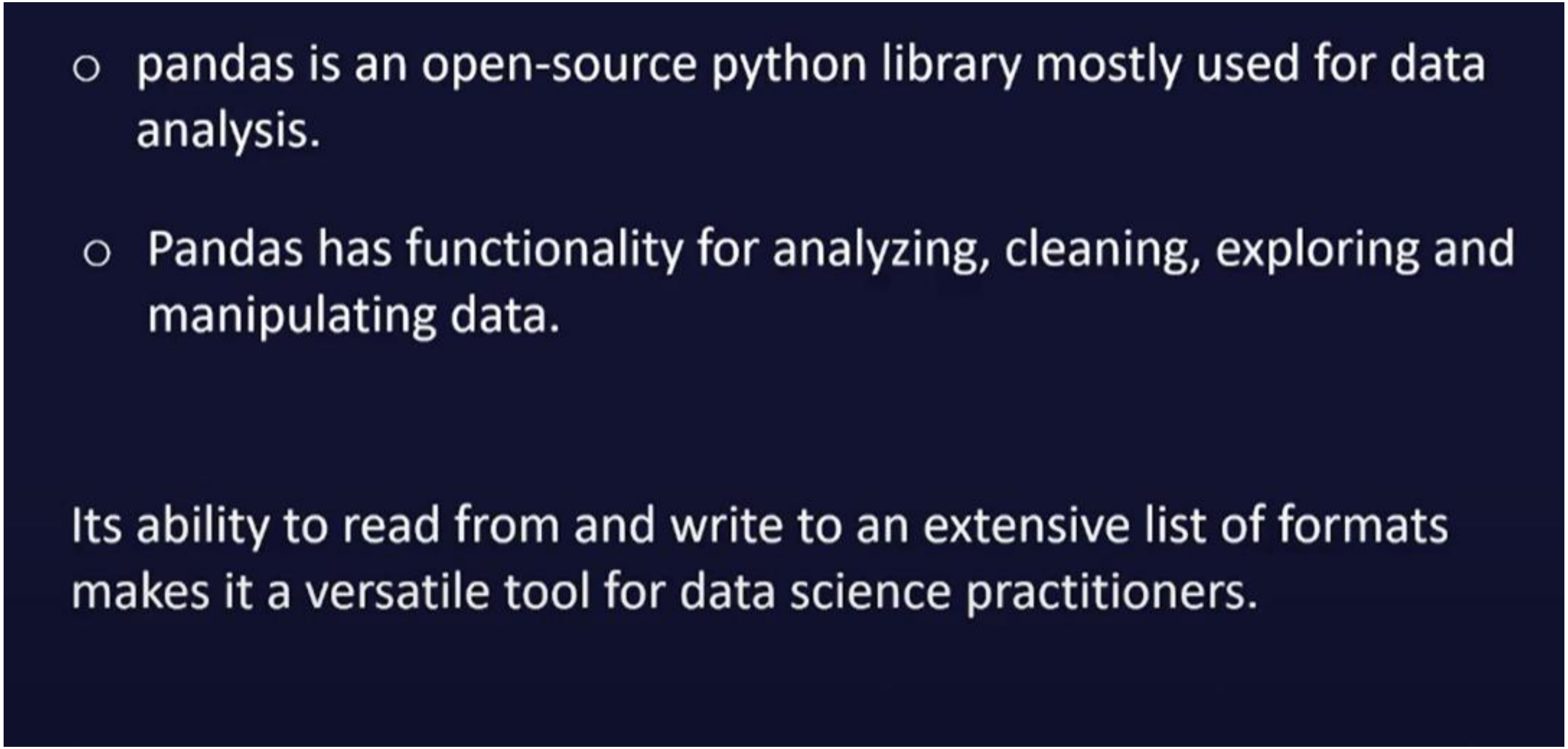


6) Machine Learning:-

- Understanding the basics of machine learning can enhance your data analysis skills.

supervised learning algorithms and unsupervised learning algorithms needs to study

PANDAS



- pandas is an open-source python library mostly used for data analysis.
- Pandas has functionality for analyzing, cleaning, exploring and manipulating data.

Its ability to read from and write to an extensive list of formats makes it a versatile tool for data science practitioners.

Excel Vs Pandas

Excel Strengths:

•**User-Friendly Interface:** Excel offers a familiar and intuitive interface for data visualization, basic data manipulation, and spreadsheet management.

•**Wide Adoption:** Excel is widely used across various industries, making collaboration and data sharing easier.

•**Formatting and Presentation:** Excel excels in data formatting, creating reports, and presenting information visually.

Excel Vs Pandas

pandas Strengths (Where Excel Falls Short):

•**Data Analysis Powerhouse:** pandas offers powerful data structures (series ,DataFrames) and functions for cleaning, manipulating, and analyzing large datasets. Excel can struggle with complex data manipulation tasks.

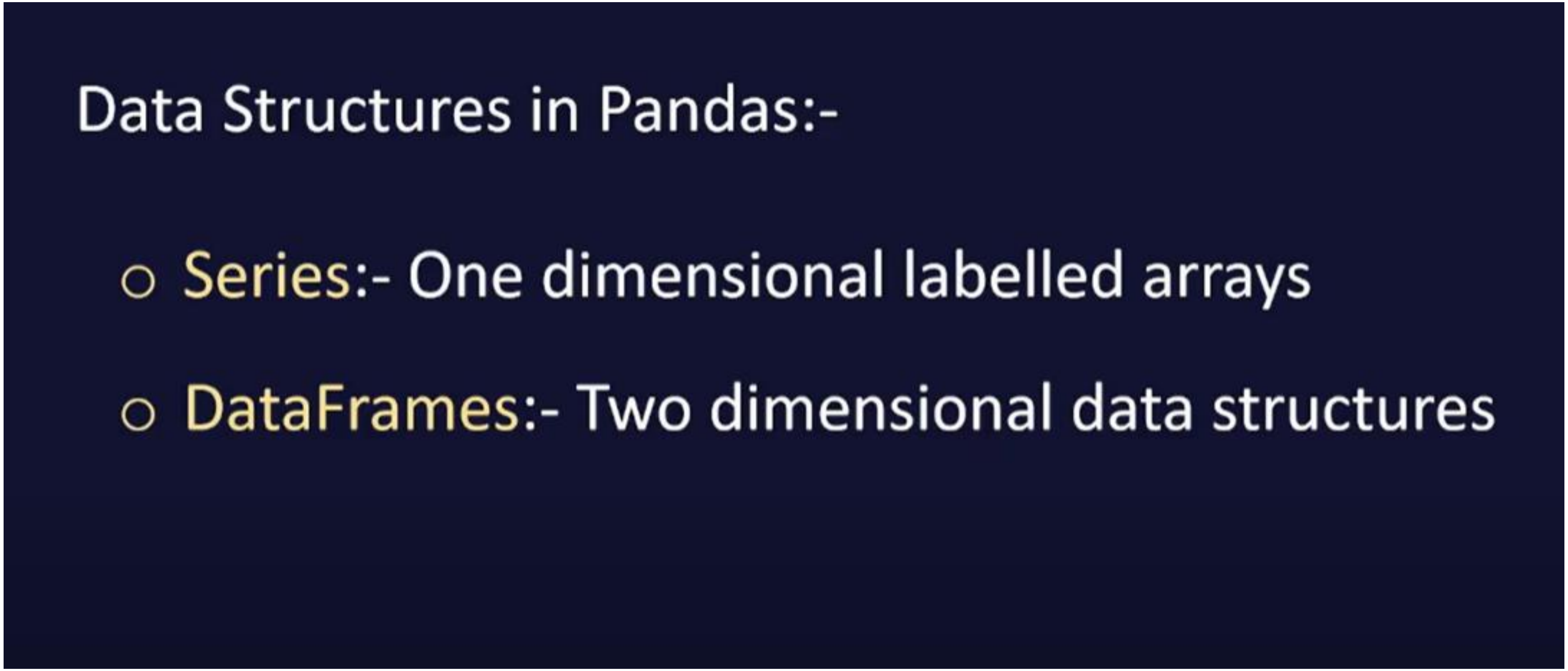
•**Scalability:** pandas can handle massive datasets efficiently, while Excel's performance can deteriorate with very large datasets.

•**Integration with Python:** pandas integrates seamlessly with the Python ecosystem, allowing for powerful scripting and automation of data analysis tasks. Excel lacks this level of programmatic control.

•**Data Cleaning and Transformation:** pandas offers robust tools for handling missing values, data type conversion, and other data cleaning tasks that can be cumbersome in Excel.

•**Merging and Joining Data:** pandas excels at merging and joining data from multiple sources, which can be complex in Excel with large datasets.

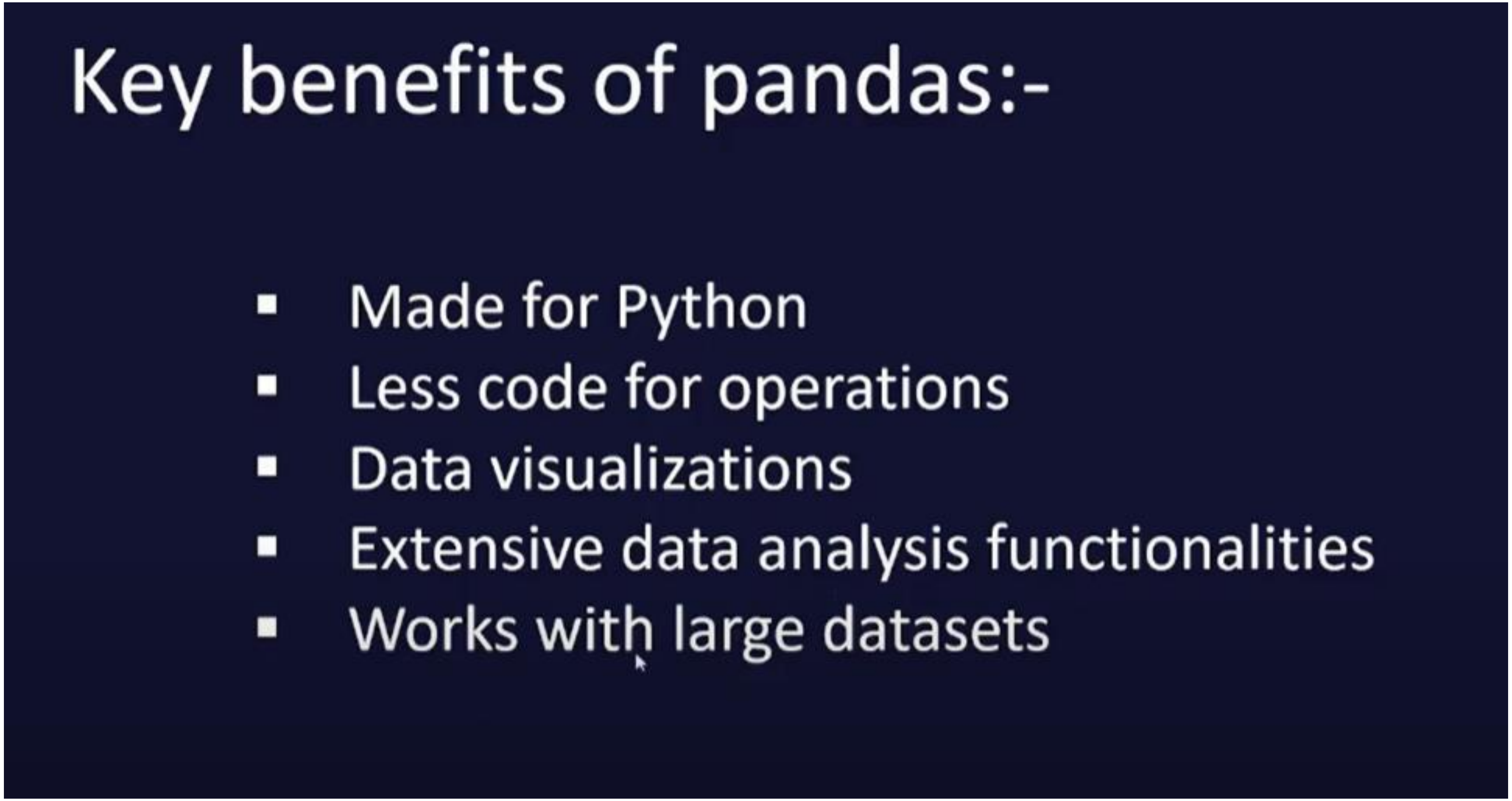
PANDAS



Data Structures in Pandas:-

- Series:- One dimensional labelled arrays
- DataFrames:- Two dimensional data structures

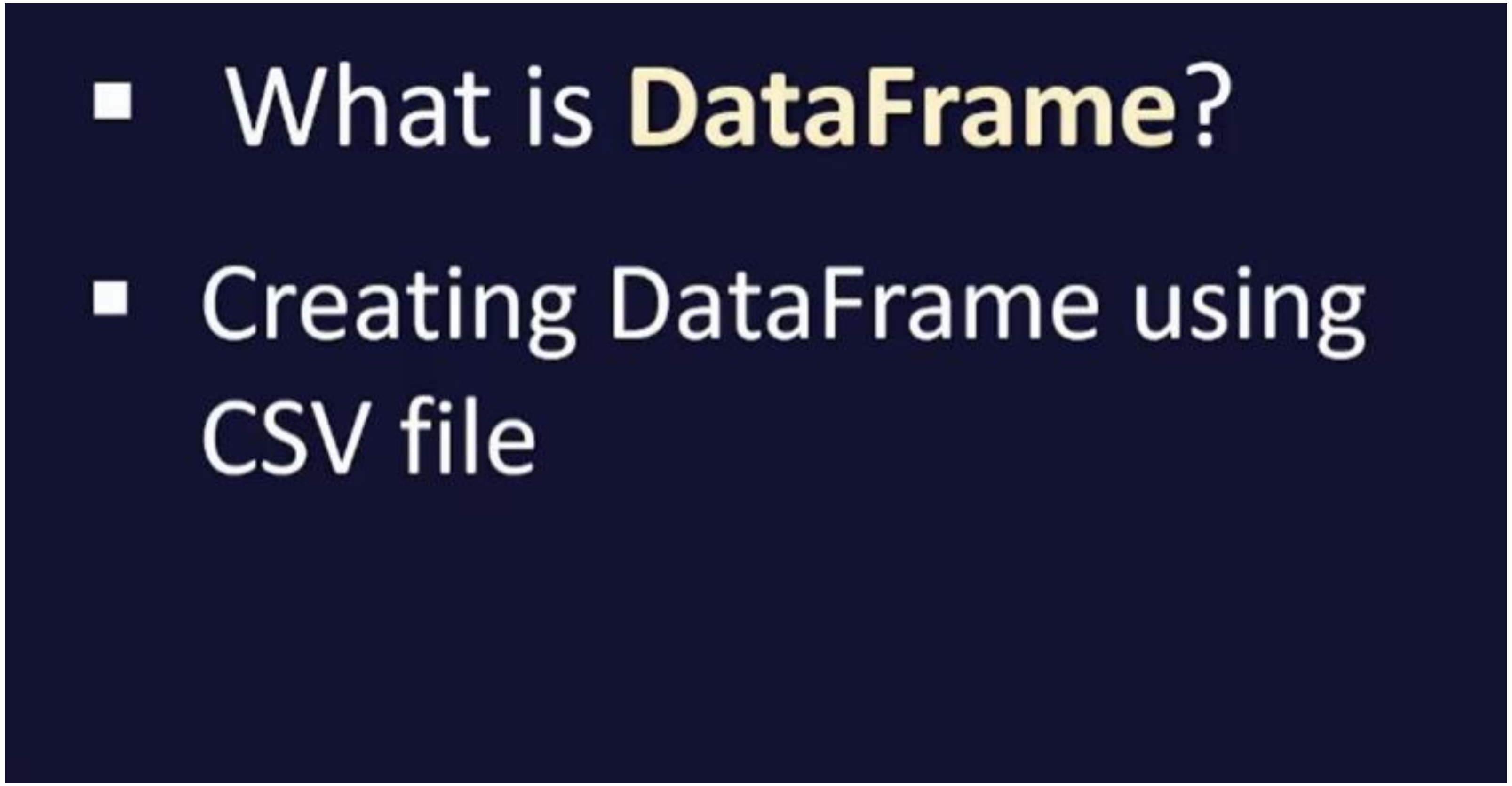
PANDAS



Key benefits of pandas:-

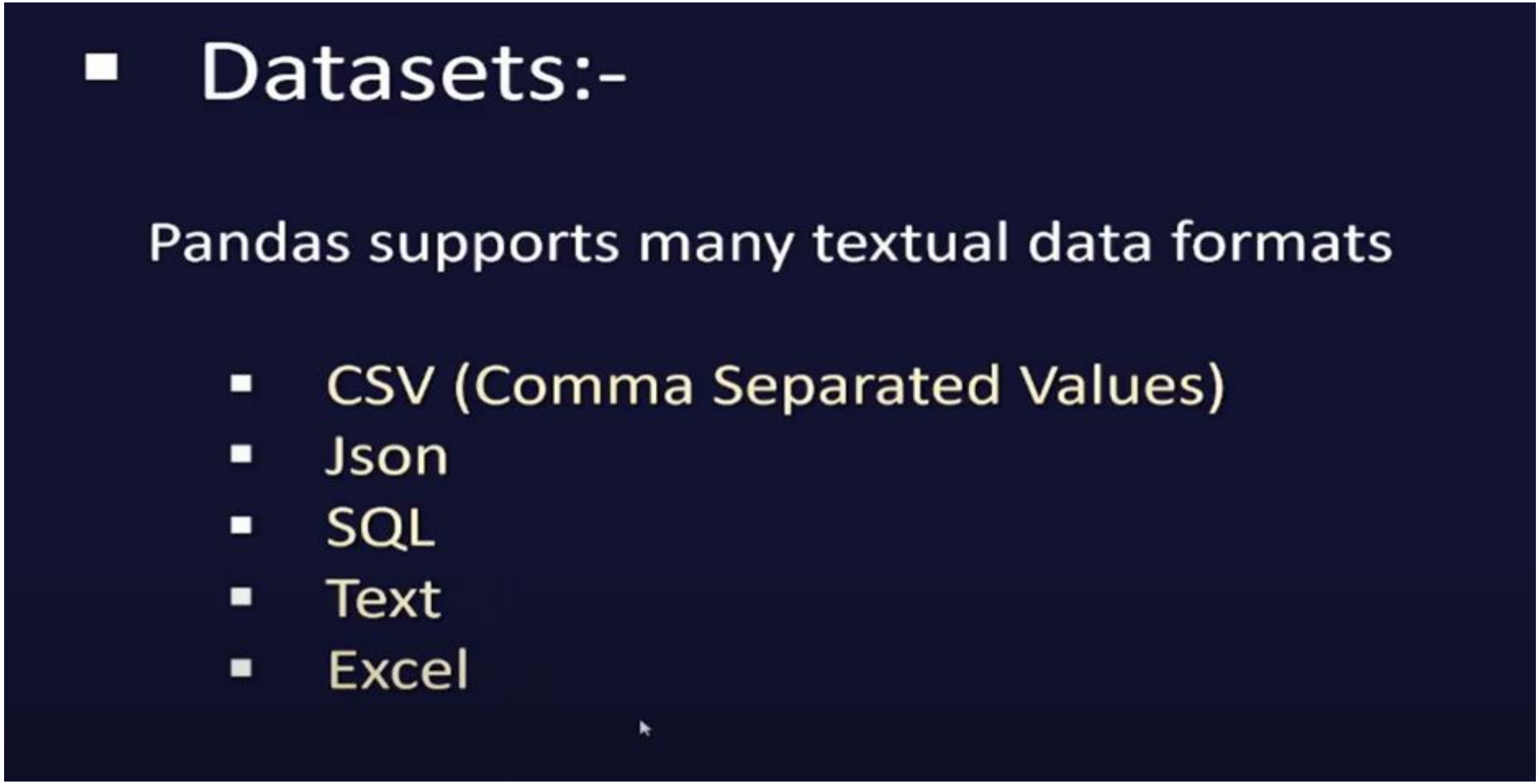
- Made for Python
- Less code for operations
- Data visualizations
- Extensive data analysis functionalities
- Works with large datasets

PANDAS



- What is **DataFrame**?
- Creating DataFrame using CSV file

PANDAS

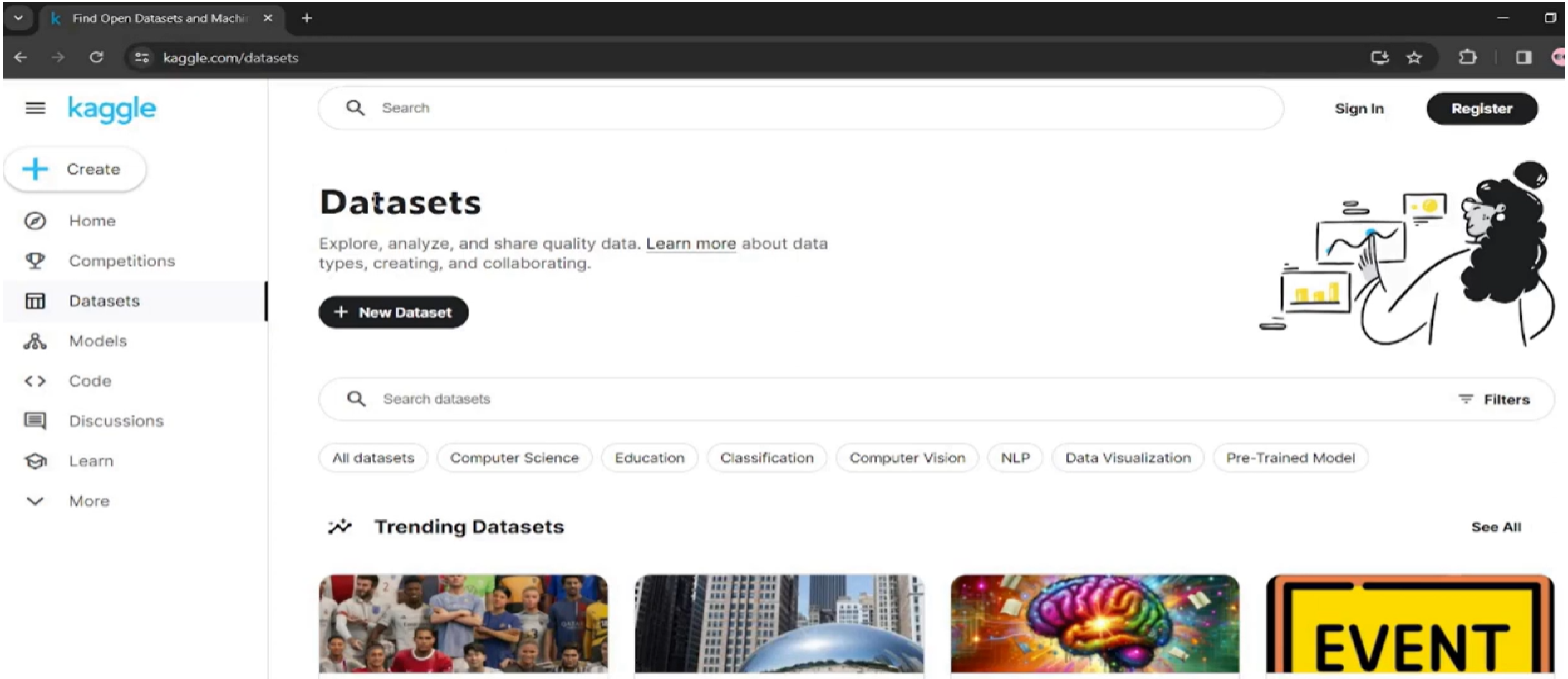


- Datasets:-

Pandas supports many textual data formats

- CSV (Comma Separated Values)
- Json
- SQL
- Text
- Excel

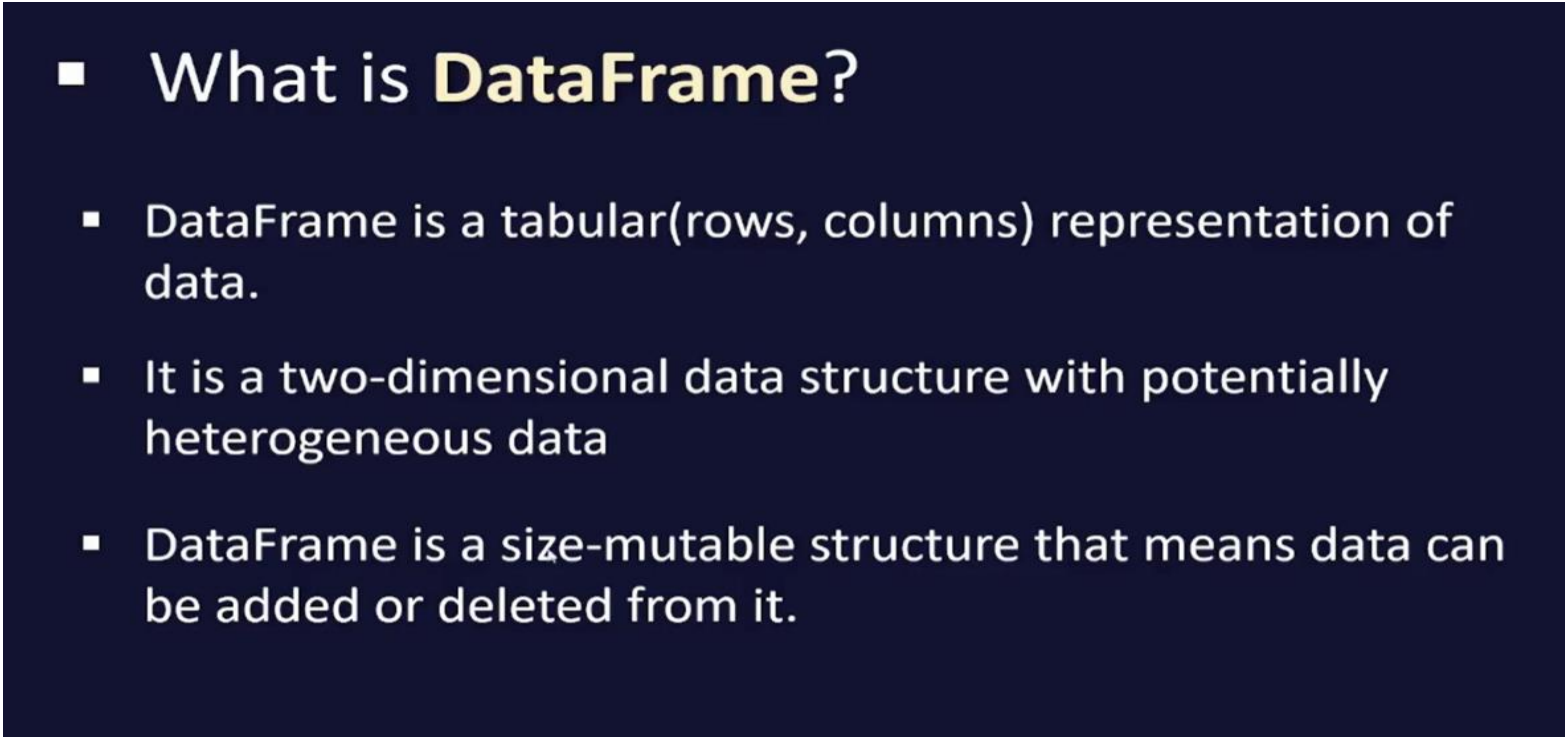
PANDAS



Datasets

Explore, analyze, and share quality data. Learn more about data types, creating, and collaborating.

PANDAS



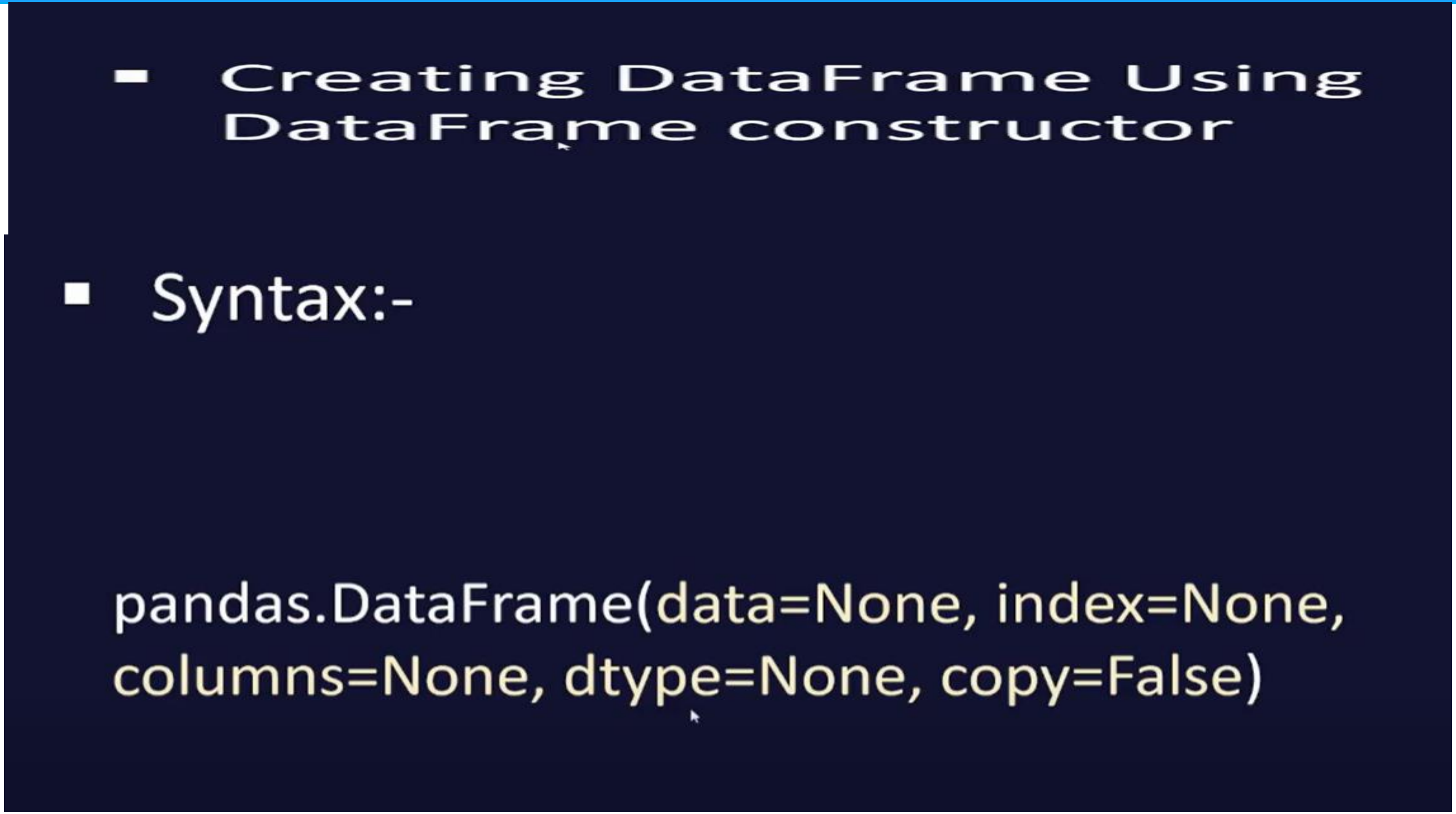
- What is DataFrame?
  - DataFrame is a tabular(rows, columns) representation of data.
  - It is a two-dimensional data structure with potentially heterogeneous data
  - DataFrame is a size-mutable structure that means data can be added or deleted from it.

PANDAS

- Creating DataFrame Using DataFrame constructor

Pandas

PANDAS

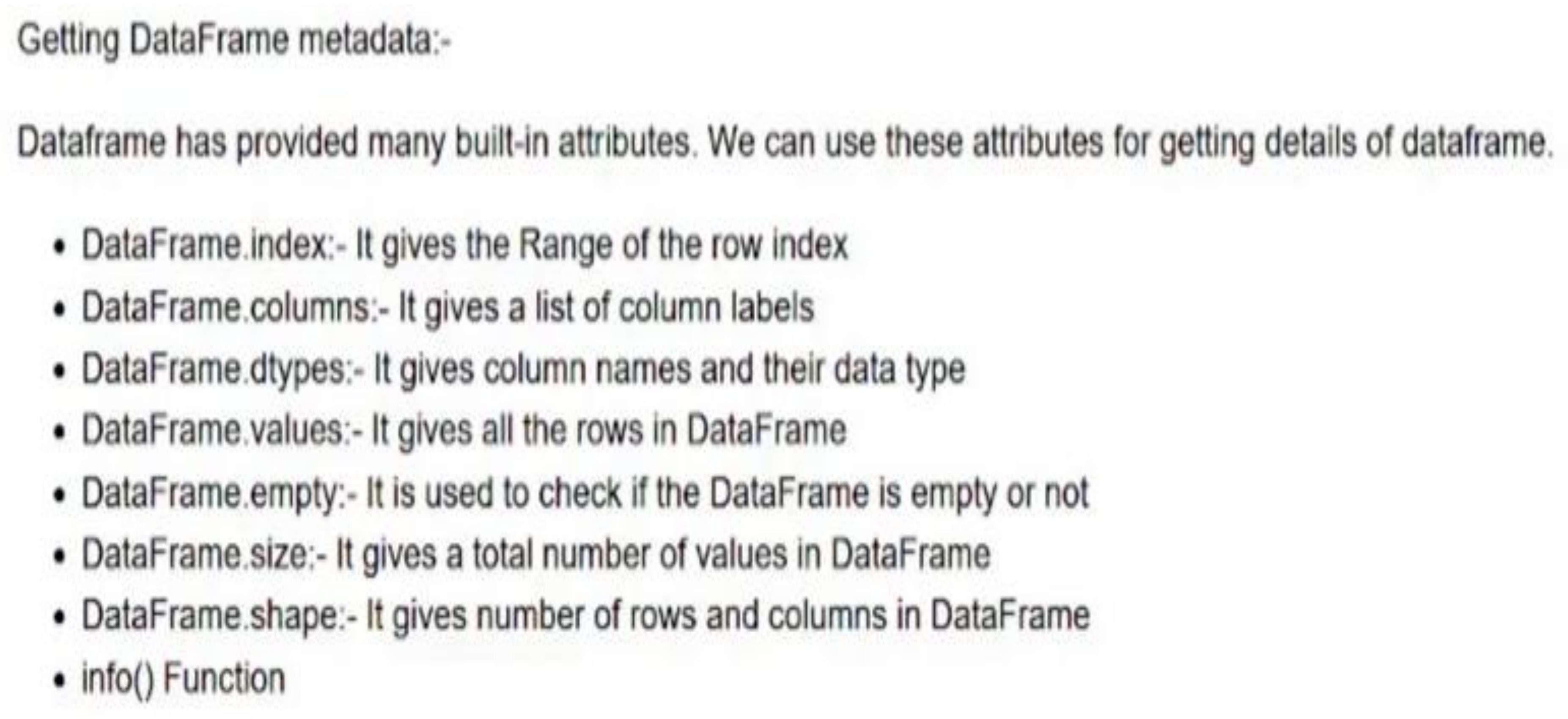


- Creating DataFrame Using DataFrame constructor

- Syntax:-

```
pandas.DataFrame(data=None, index=None,
columns=None, dtype=None, copy=False)
```

PANDAS

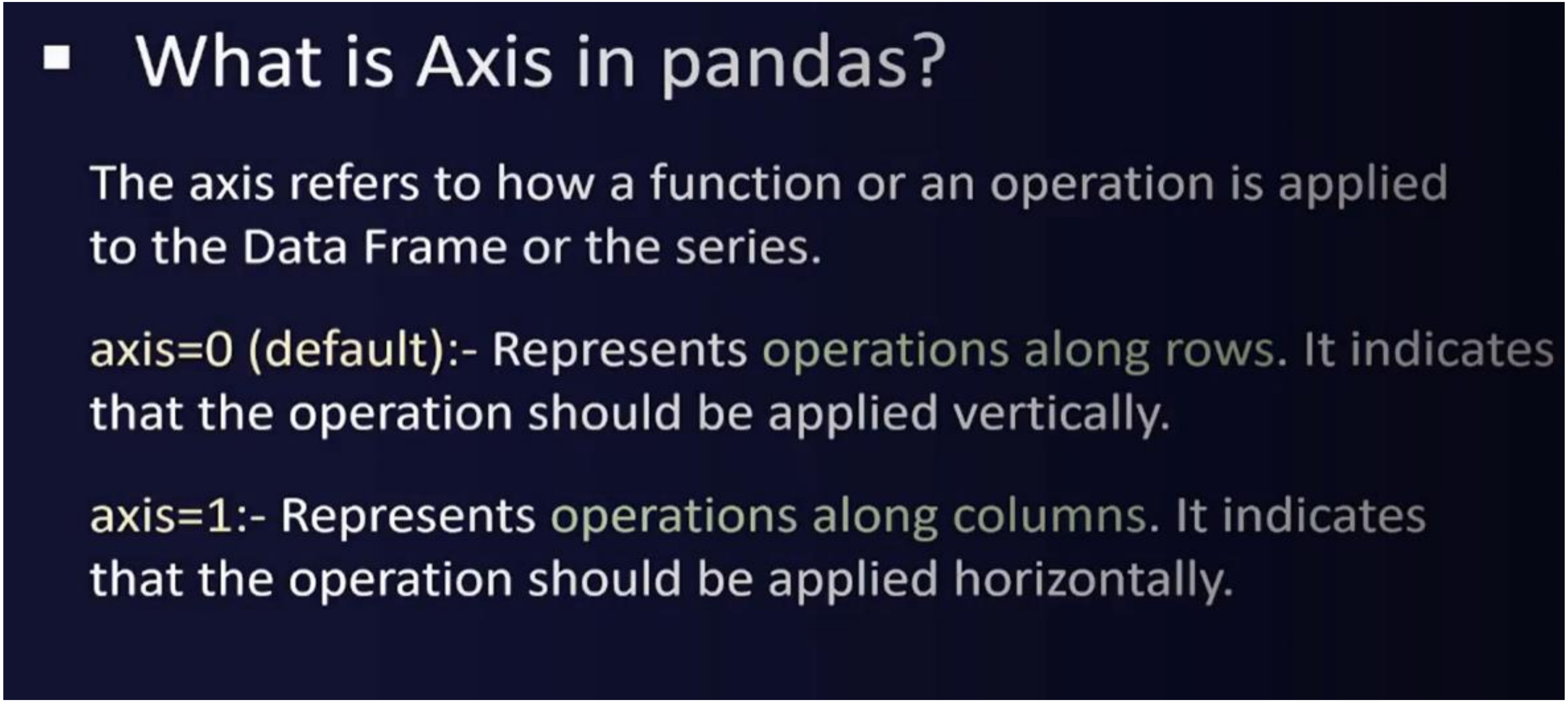


Getting DataFrame metadata:-

Dataframe has provided many built-in attributes. We can use these attributes for getting details of dataframe.

- DataFrame.index:- It gives the Range of the row index
- DataFrame.columns:- It gives a list of column labels
- DataFrame.dtypes:- It gives column names and their data type
- DataFrame.values:- It gives all the rows in DataFrame
- DataFrame.empty:- It is used to check if the DataFrame is empty or not
- DataFrame.size:- It gives a total number of values in DataFrame
- DataFrame.shape:- It gives number of rows and columns in DataFrame
- info() Function

PANDAS



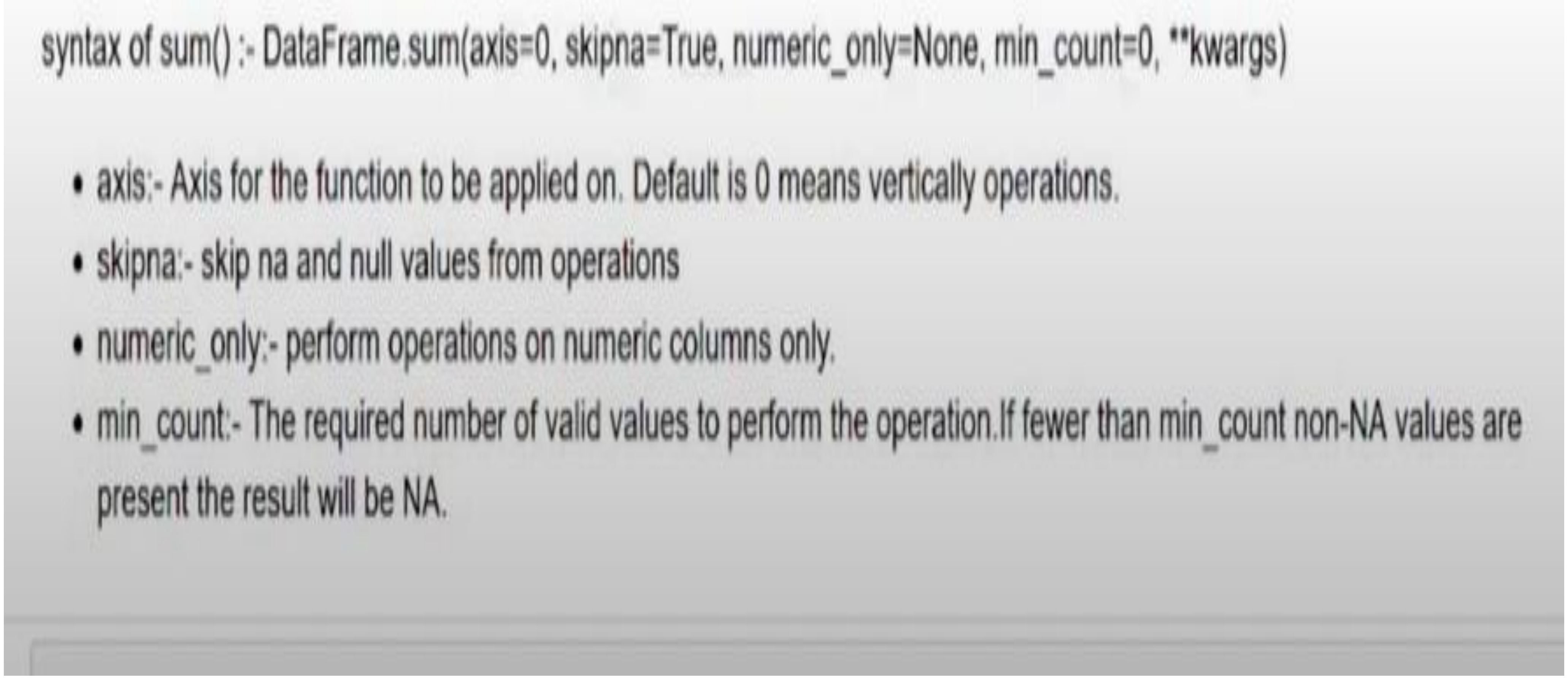
- What is Axis in pandas?

The axis refers to how a function or an operation is applied to the Data Frame or the series.

axis=0 (default):- Represents operations along rows. It indicates that the operation should be applied vertically.

axis=1:- Represents operations along columns. It indicates that the operation should be applied horizontally.

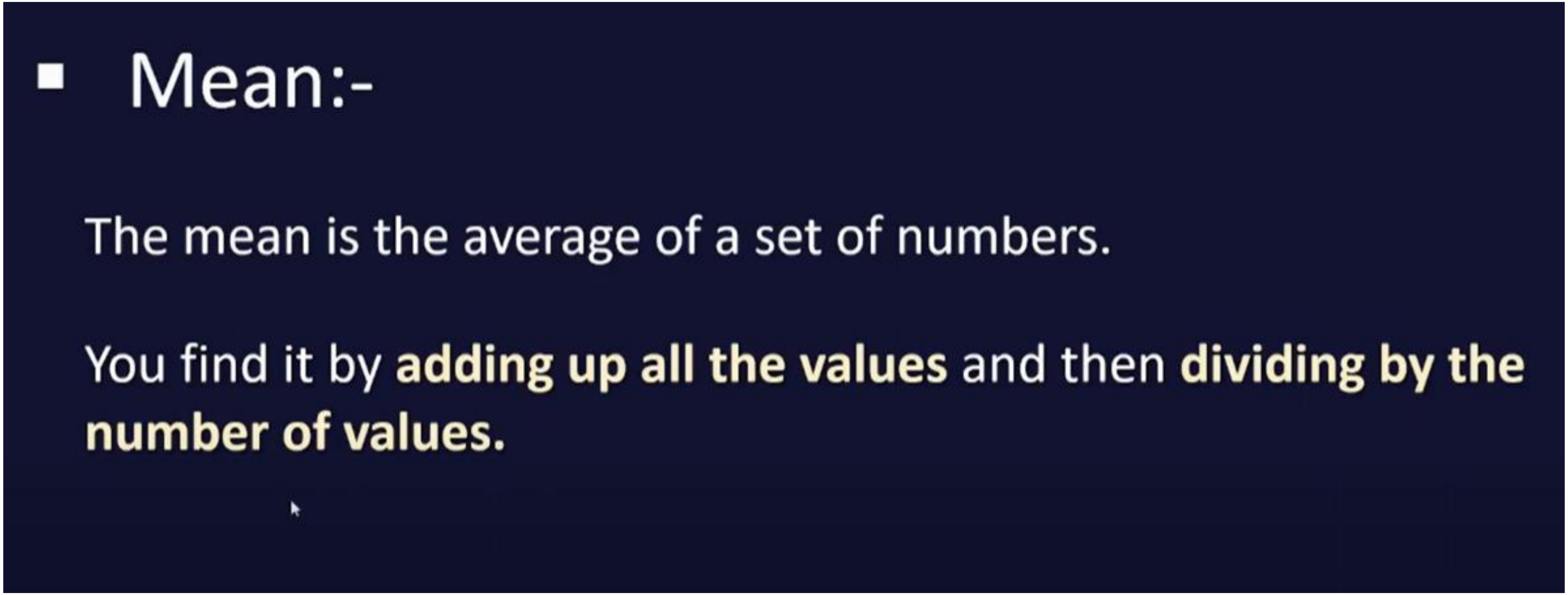
PANDAS- SUM()



syntax of sum() :- DataFrame.sum(axis=0, skipna=True, numeric_only=None, min_count=0, **kwargs)

- axis:- Axis for the function to be applied on. Default is 0 means vertically operations.
- skipna:- skip na and null values from operations
- numeric_only:- perform operations on numeric columns only.
- min_count:- The required number of valid values to perform the operation.If fewer than min_count non-NA values are present the result will be NA.

PANDAS- SUM()

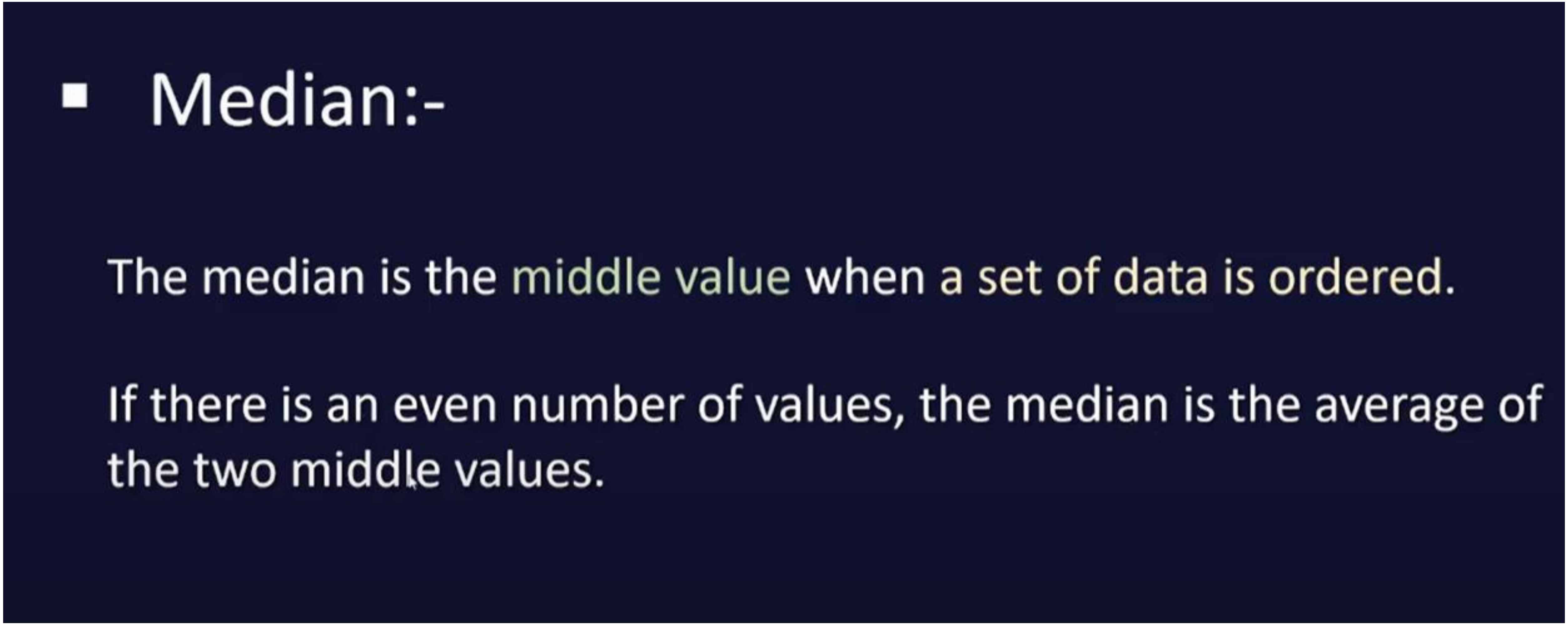


- Mean:-

The mean is the average of a set of numbers.

You find it by **adding up all the values** and then **dividing by the number of values.**

PANDAS- SUM()

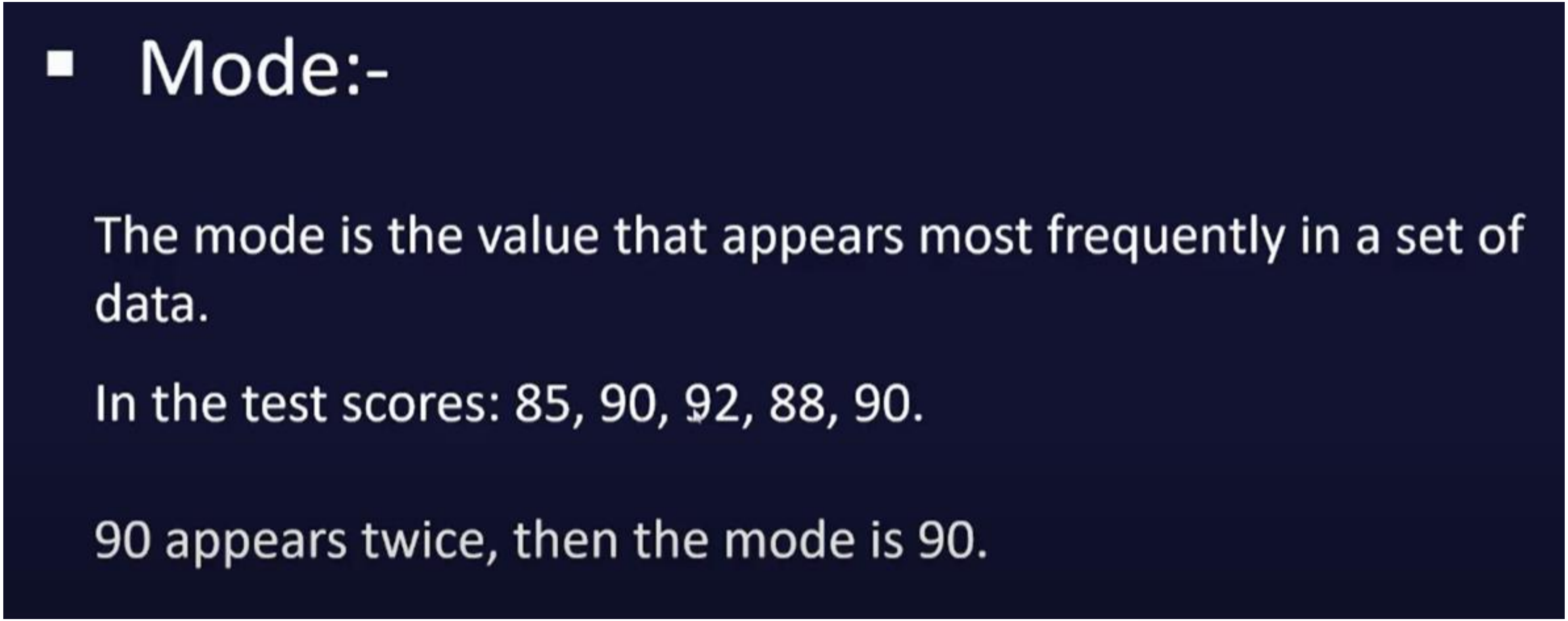


- Median:-

The median is the middle value when a set of data is ordered.

If there is an even number of values, the median is the average of the two middle values.

PANDAS- SUM()



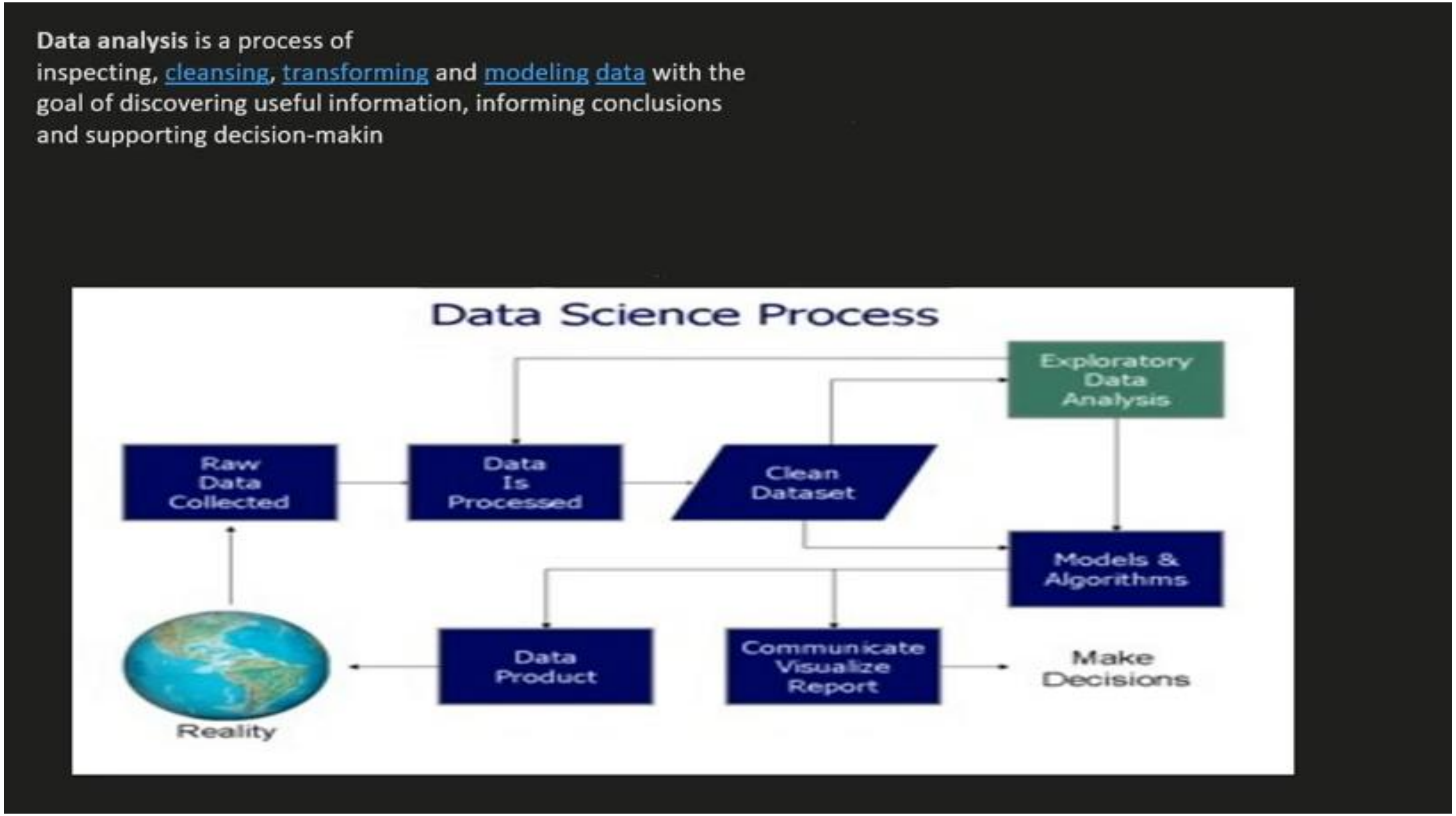
- Mode:-

The mode is the value that appears most frequently in a set of data.

In the test scores: 85, 90, 92, 88, 90.

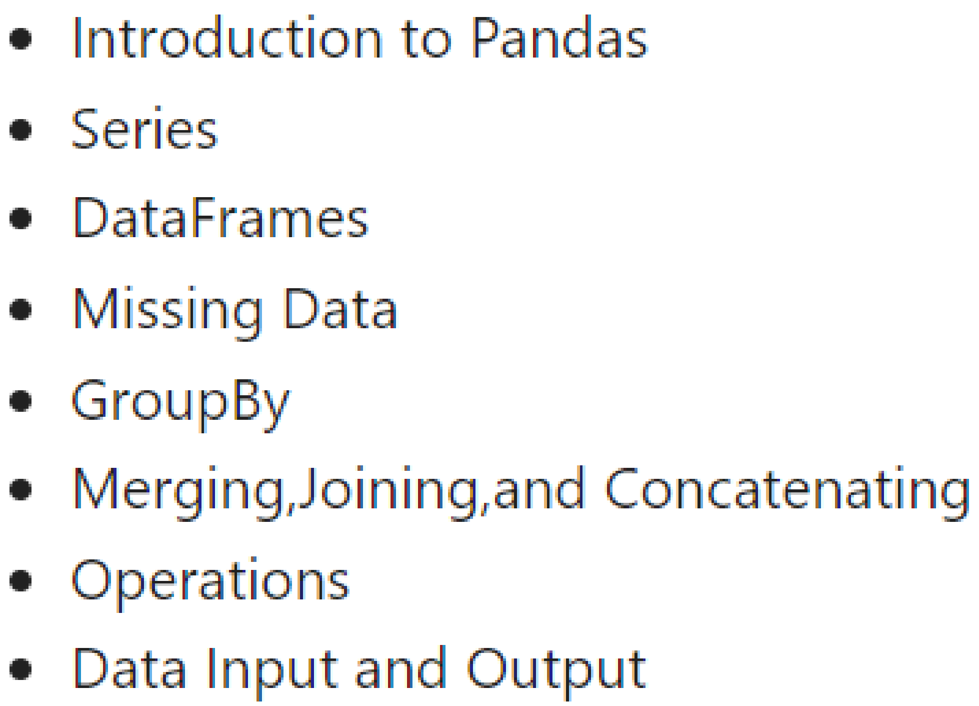
90 appears twice, then the mode is 90.

Data Analysis



Data analysis is a process of inspecting, cleansing, transforming and modeling data with the goal of discovering useful information, informing conclusions and supporting decision-makin

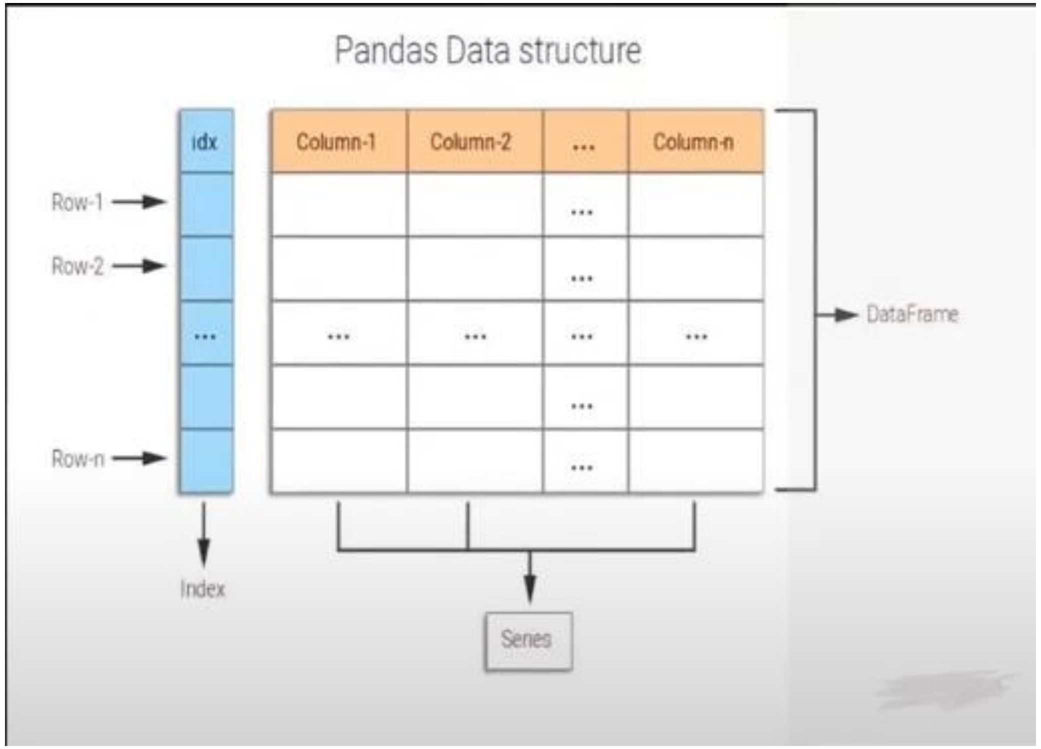
Pandas



- Introduction to Pandas
- Series
- DataFrames
- Missing Data
- GroupBy
- Merging,Joining,and Concatenating
- Operations
- Data Input and Output

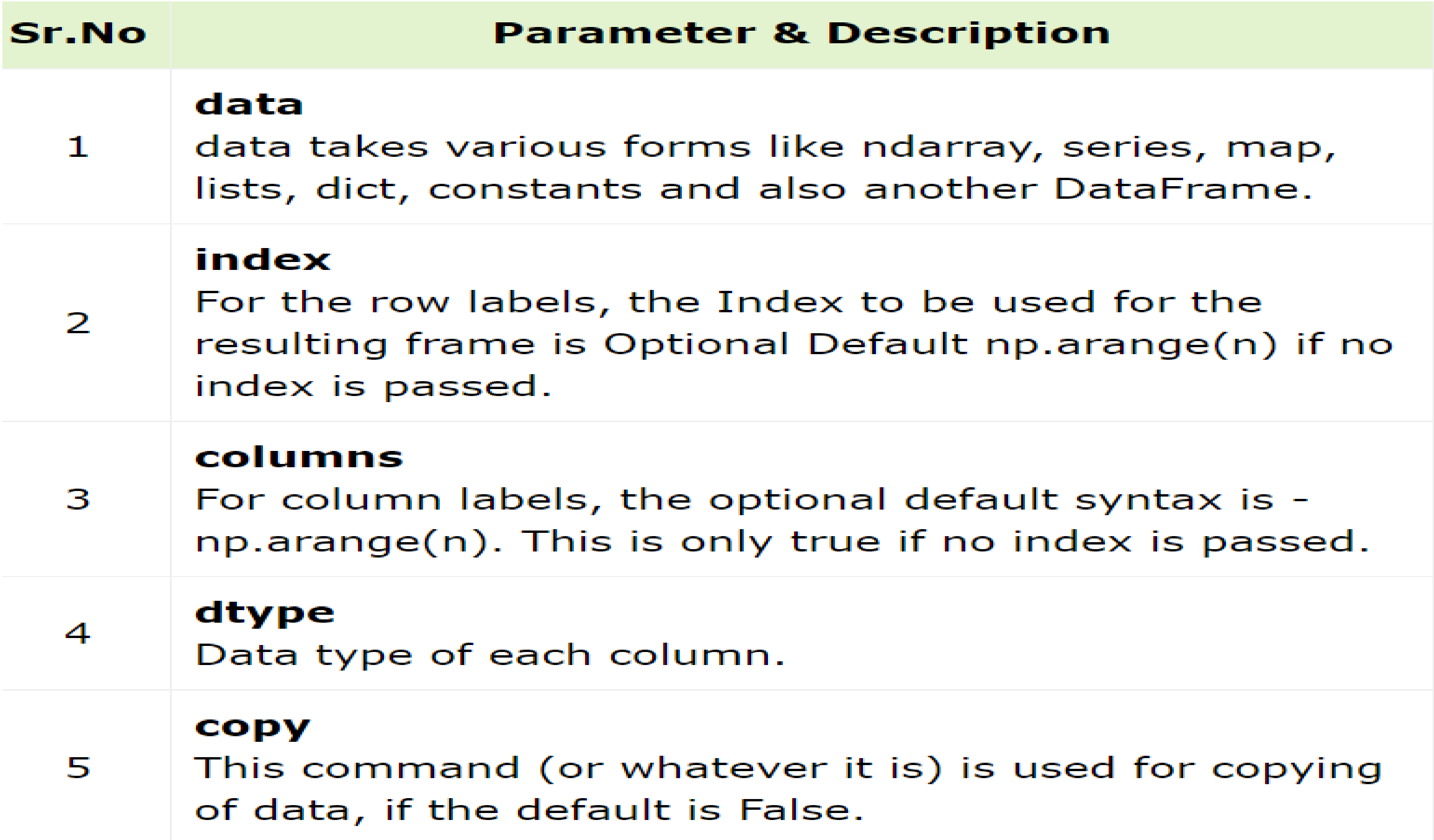
Pandas Data Structure

.



Pandas Data Structure

pandas.DataFrame(data, index, columns, dtype, copy)



Sr.No	Parameter & Description
1	**data** data takes various forms like ndarray, series, map, lists, dict, constants and also another DataFrame.
2	**index** For the row labels, the Index to be used for the resulting frame is Optional Default np.arange(n) if no index is passed.
3	**columns** For column labels, the optional default syntax is - np.arange(n). This is only true if no index is passed.
4	**dtype** Data type of each column.
5	**copy** This command (or whatever it is) is used for copying of data, if the default is False.