

# DATA SCIENCE

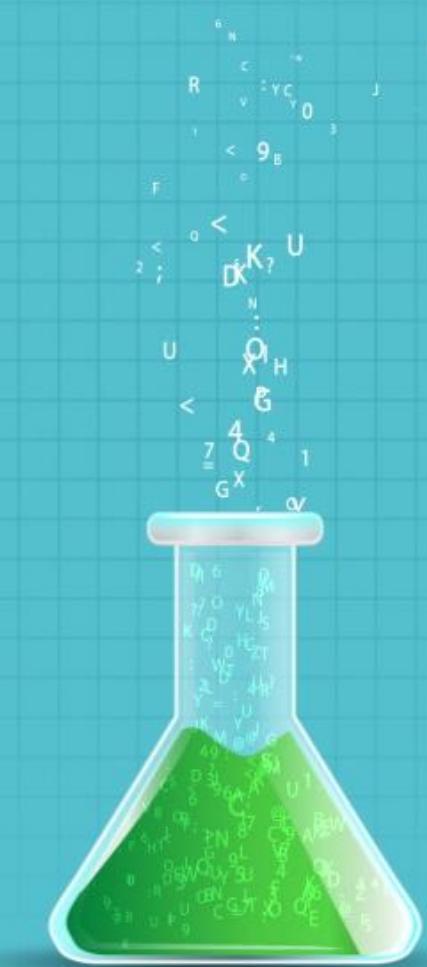
## Data Science with Python

### Lesson 1—Data Science Overview



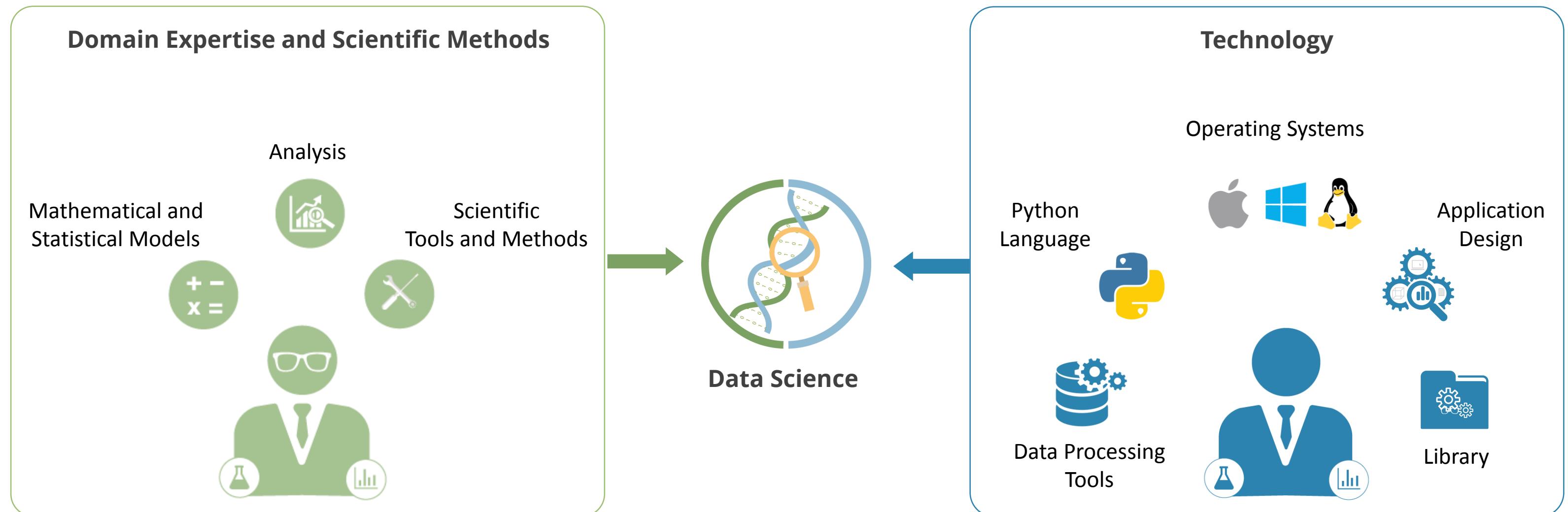
# What You'll Learn

- Know what Data Science is
- Discuss the roles and responsibilities of a Data Scientist
- List various applications of Data Science
- Understand how Data Science and Big Data work together
- Explore Data Science as a discipline
- Understand how and why Data Science is gaining importance
- Understand what Python is and what problems it resolves



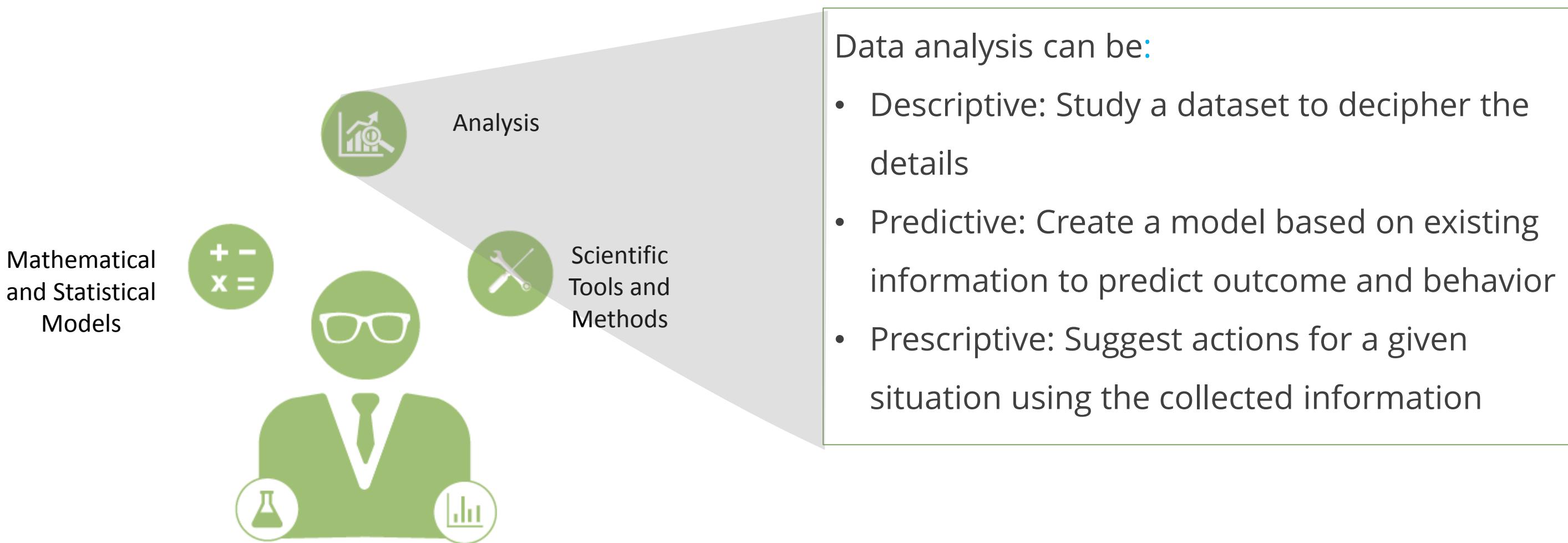
# The Components of Data Science

When we combine domain expertise and scientific methods with technology, we get Data Science.



# Domain Expertise and Scientific Methods

Data Scientists collect data and explore, analyze, and visualize it. They apply mathematical and statistical models to find patterns and solutions in the data.

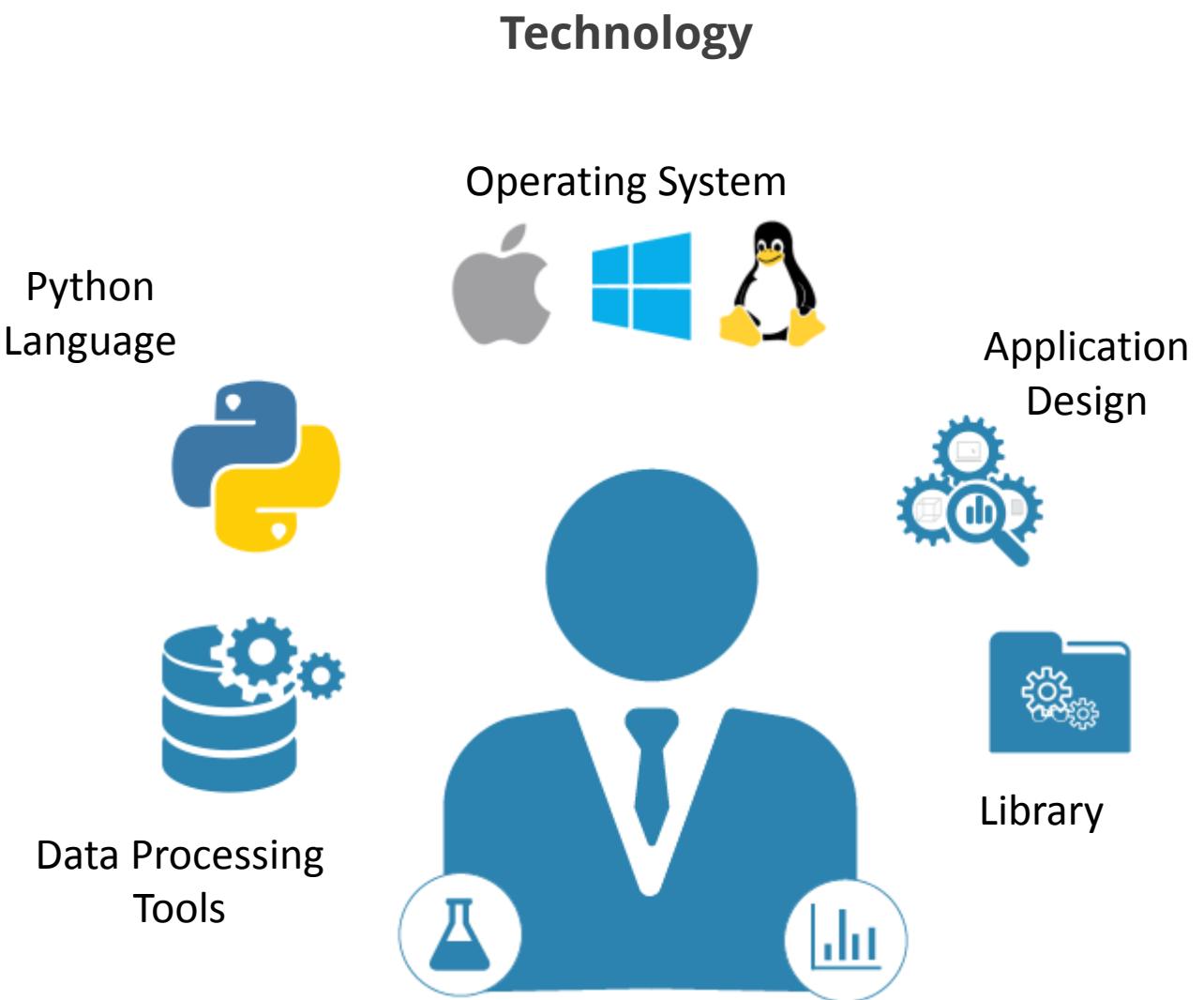


# Data Processing and Analytics

Modern tools and technologies have made data processing and analytics faster and efficient.

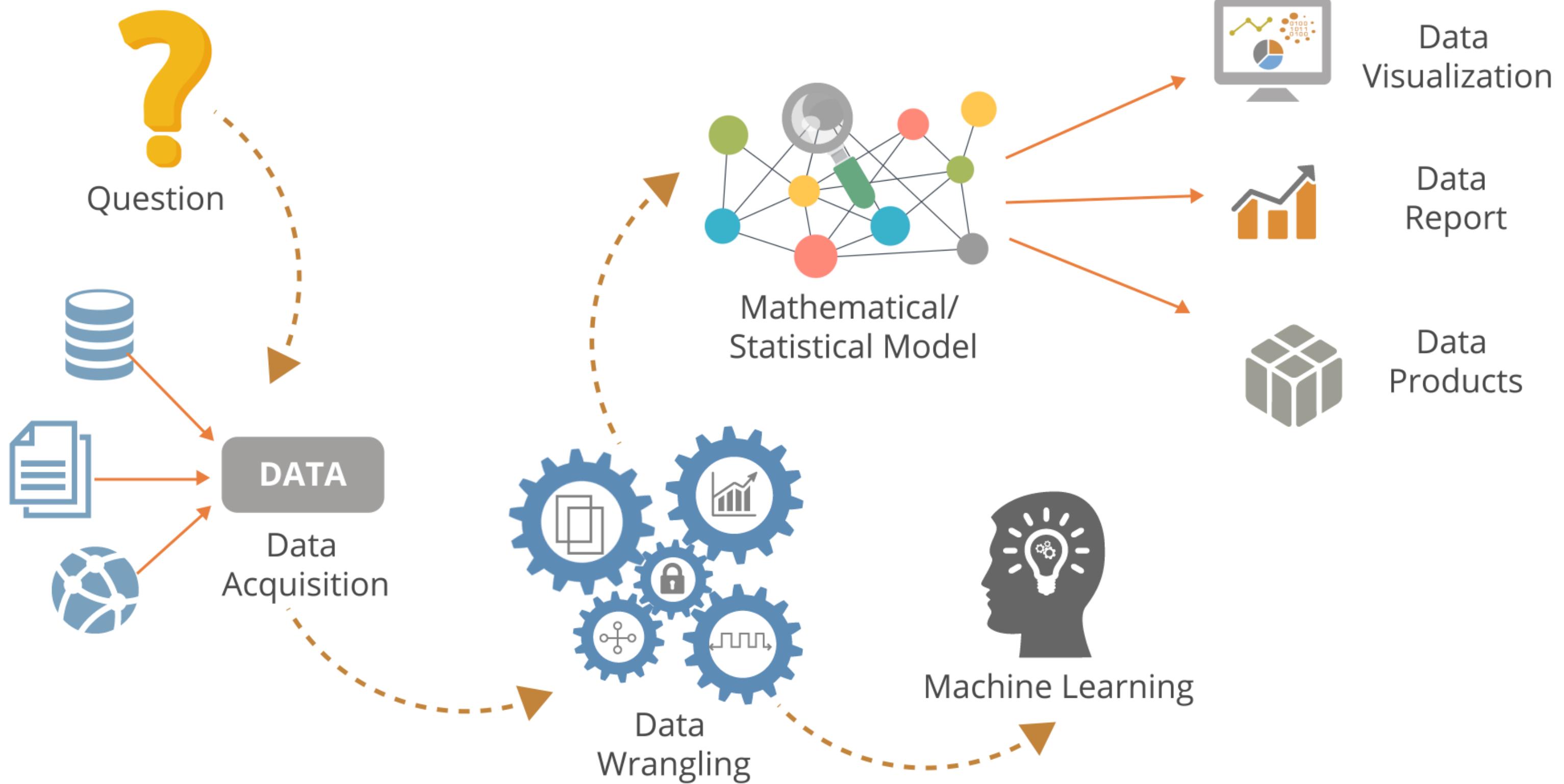
These technologies help Data Scientists to

- Build and train machine learning models
- Manipulate data with technology
- Build data tools, applications, and services
- Extract information from data



Data analysis that uses only technology and domain knowledge without mathematical and statistical knowledge often leads to incorrect patterns and wrong interpretations. This can cause serious damage to businesses.

# A Day in a Data Scientist's Life



# Basic Skills of a Data Scientist

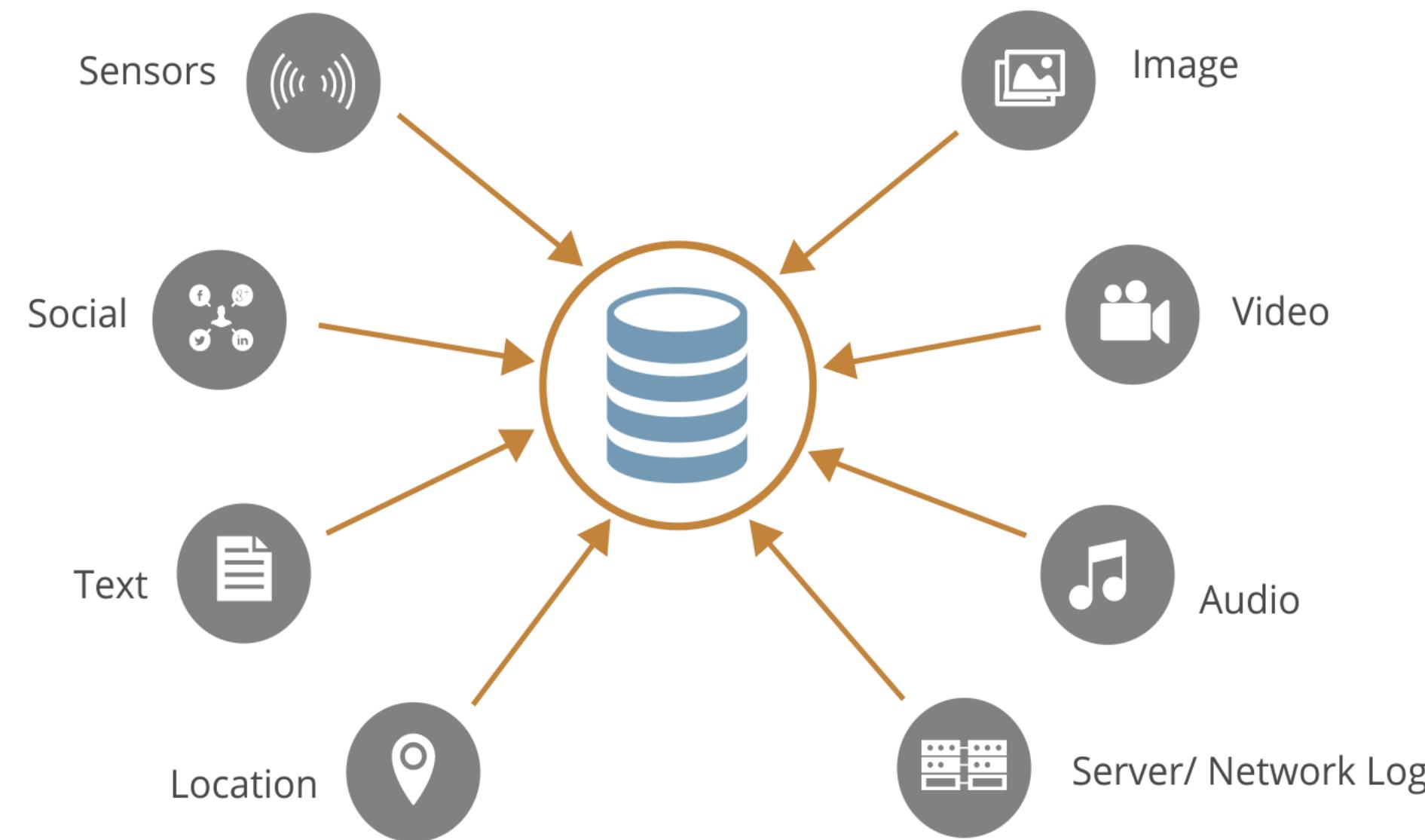
A Data Scientist should be able to

- Ask the right questions
- Understand data structure
- Interpret and wrangle data
- Apply statistical and mathematical methods
- Visualize data and communicate with stakeholders
- Work as a team player



# Sources of Big Data

Data Scientists work with different types of datasets for various purposes. Now that Big Data is generated every second through different media, the role of Data Science has become more important.



# The 3 Vs of Big Data

Big Data is characterized by the following:

Volume

Enormous amount of data generated from various sources

Velocity

Large amount of data streaming in at great speeds, which requires quick data processing

Variety

Different formats of data: Structured, Semi-structured, and Unstructured

Big Data is a huge collection of data stored on distributed systems/machines popularly referred to as Hadoop clusters. Data Science helps extract information from the Data and build information-driven enterprises.

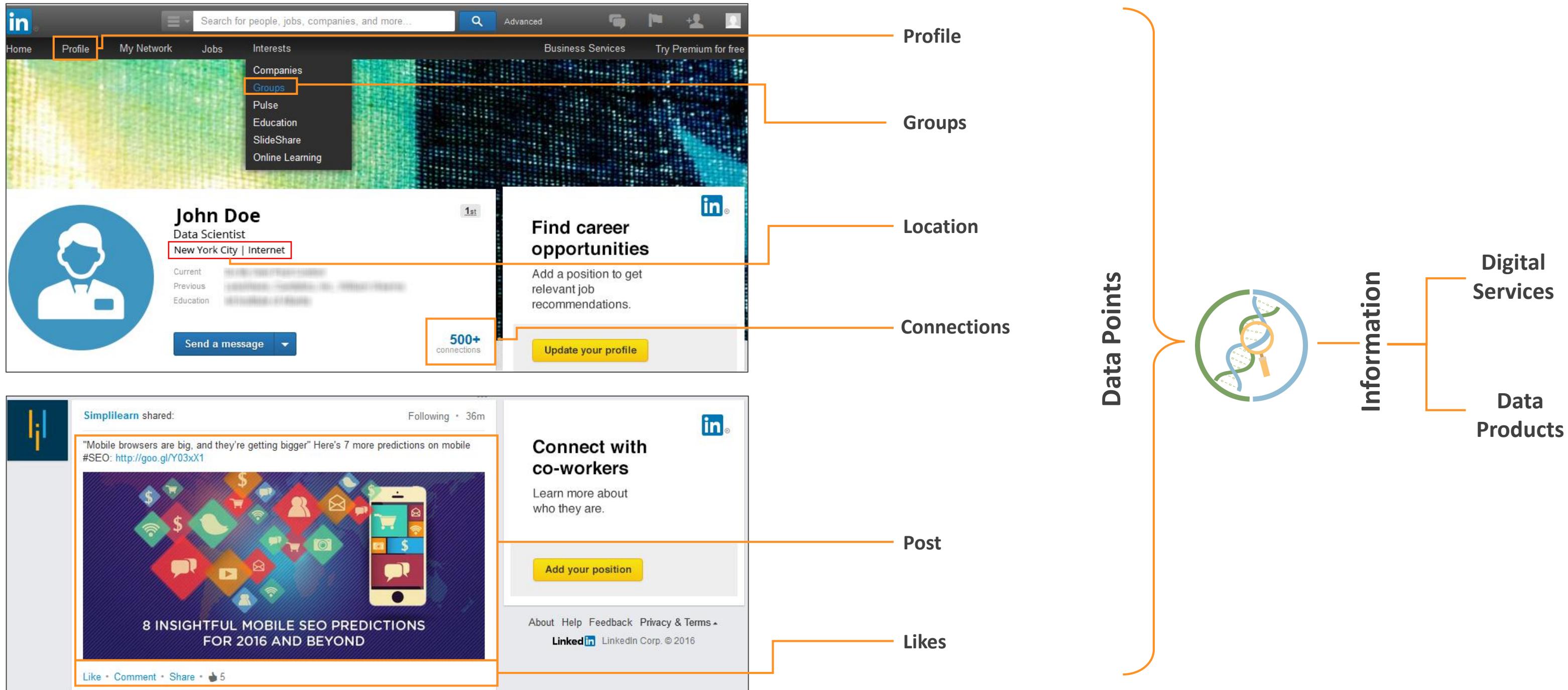
# Different Sectors Using Data Science

Various sectors use Data Science to extract the information they need to create different services and products.



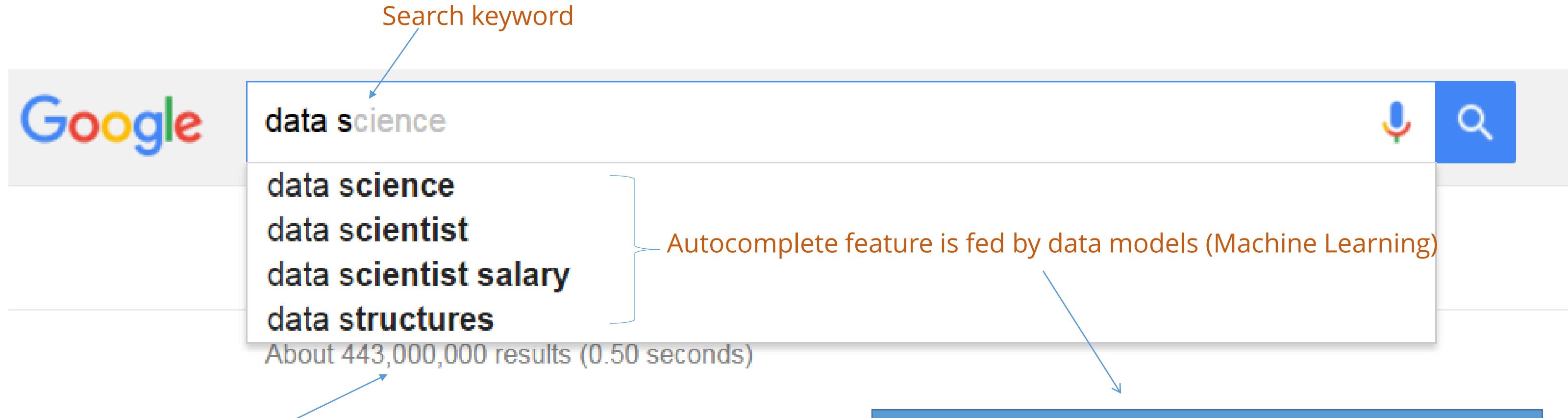
# Using Data Science—Social Network Platforms

LinkedIn uses data points from its users to provide them with relevant digital services and data products.



# Using Data Science—Search Engines

Google uses Data Science to provide relevant search recommendations as the user types a query.



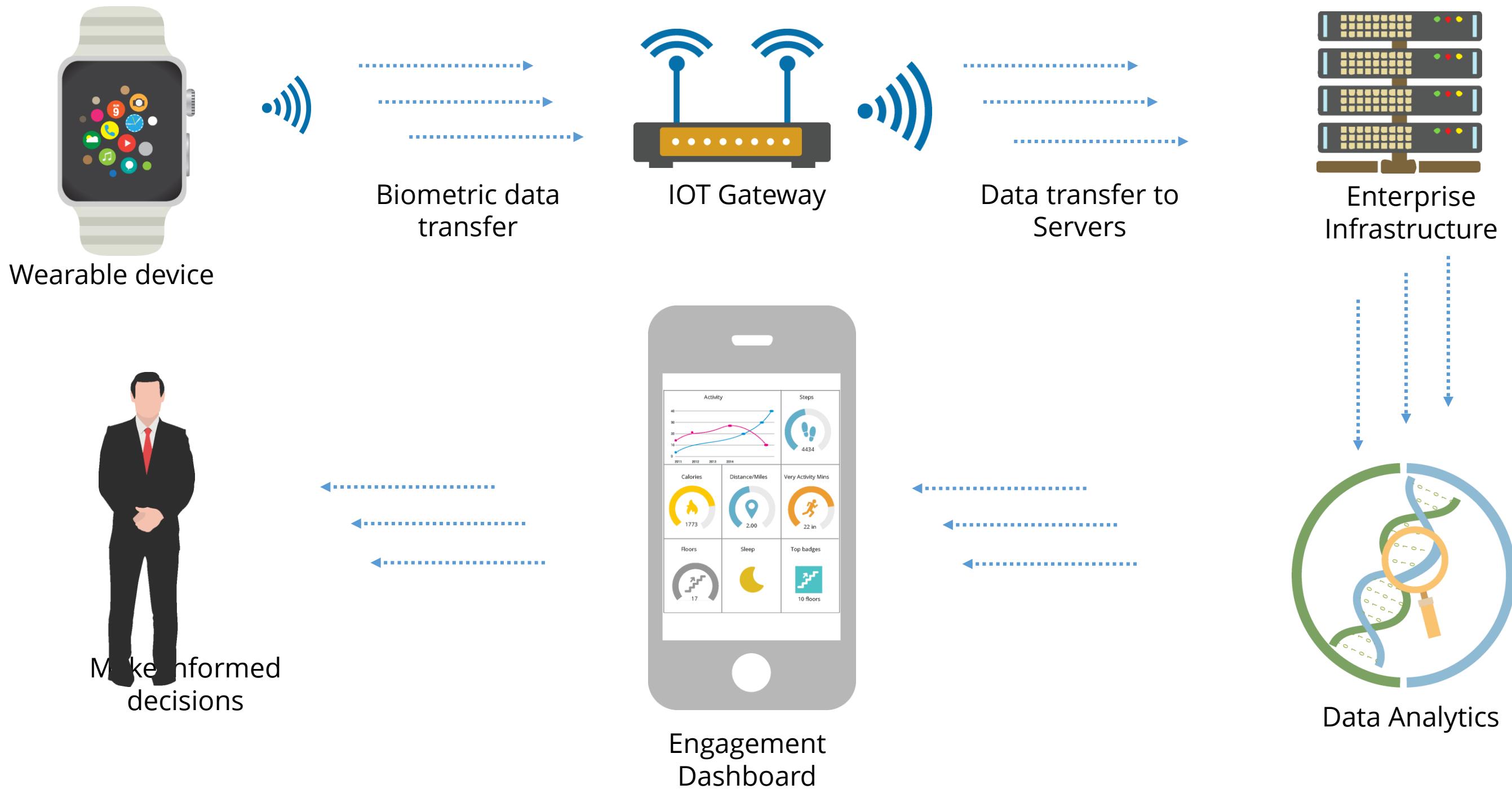
Fast and real-time analytics is made possible by modern and advanced infrastructure, tools, and technologies

## Influencing Factors

1. Query Volume – Unique and verifiable users
2. Geographical locations
3. Keyword/phrase matches on the web
4. Some scrubbing for inappropriate content

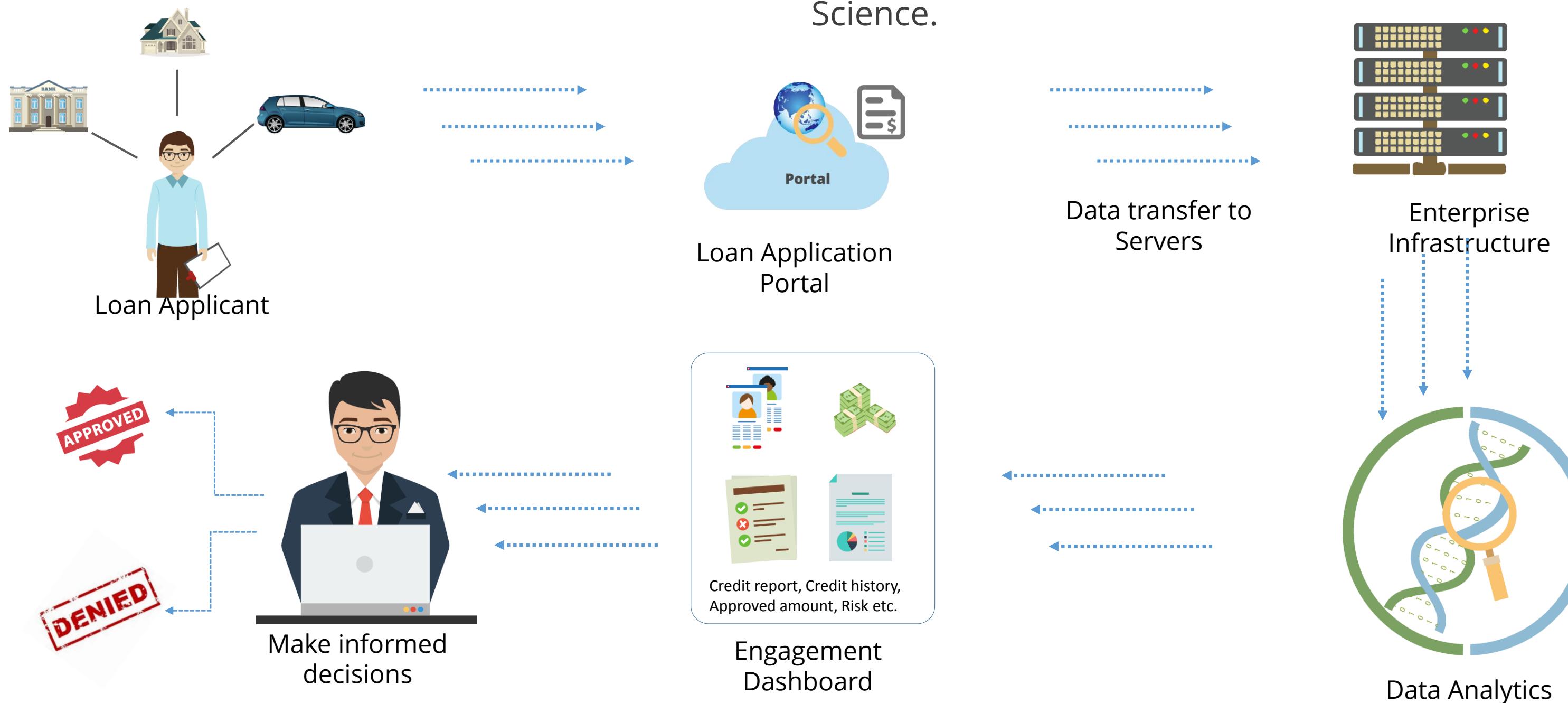
# Using Data Science—Healthcare

Wearable devices use Data Science to analyze data gathered by their biometric sensors.



# Using Data Science—Finance

A Loan Manager can easily access and sift through a loan applicant's financial details using Data Science.



# Using Data Science—Public Sector

The governments in different countries share large datasets from various domains with the public.

Data.gov is a website hosted and maintained by the U.S. government.

The screenshot shows the Data.gov homepage. At the top, there is a navigation bar with links for DATA, TOPICS, IMPACT, APPLICATIONS, DEVELOPERS, and CONTACT. Below this, a blue header banner reads "The home of the U.S. Government's open data" and "Here you will find data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and [more](#)". In the center, there is a "GET STARTED" button with a dropdown menu containing "SEARCH OVER 195,469 DATASETS". Below this is a search bar with the placeholder "Federal Student Loan Program Data" and a magnifying glass icon. To the right of the search bar is a "BROWSE TOPICS" section. This section contains two rows of icons and labels. The first row includes Agriculture (wheat), Business (store), Climate (sun over bars), Consumer (shopping cart), Ecosystems (globe), Education (graduation cap), and Energy (lightbulb). The second row includes Finance (coins), Health (cross), Local Government (map), Manufacturing (factory), Ocean (wave), Public Safety (warning sign), and Science & Research (test tube). A red box highlights the "BROWSE TOPICS" section, and a red arrow points from this box to the text "Sectors/Domains". Another red arrow points from the "SEARCH OVER 195,469 DATASETS" button to the text "Large collection of datasets".

DATA TOPICS IMPACT APPLICATIONS DEVELOPERS CONTACT

The home of the U.S. Government's open data

Here you will find data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and [more](#).

GET STARTED

SEARCH OVER 195,469 DATASETS

Federal Student Loan Program Data

BROWSE TOPICS

Agriculture      Business      Climate      Consumer      Ecosystems      Education      Energy

Finance      Health      Local Government      Manufacturing      Ocean      Public Safety      Science & Research

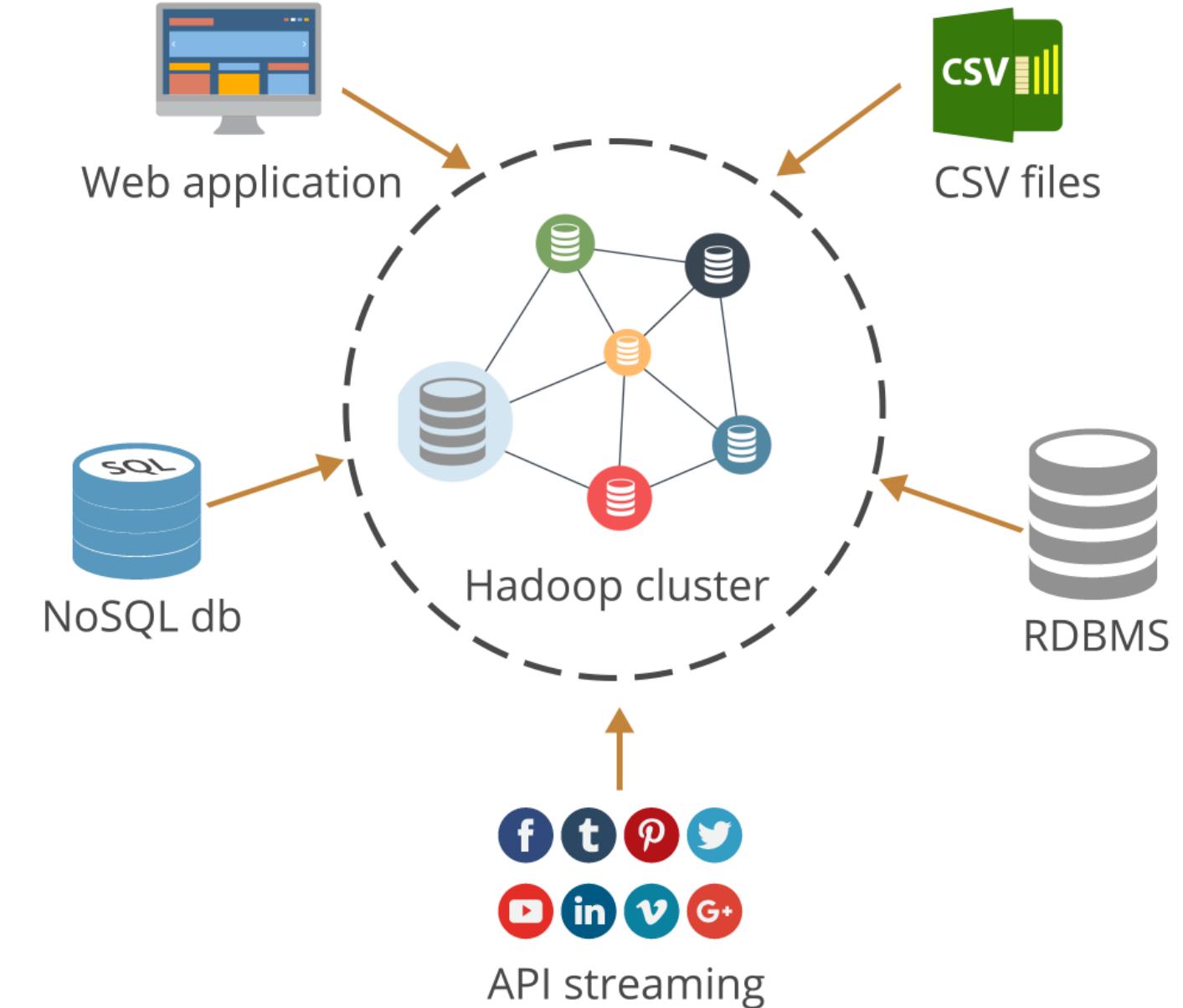
Large collection of datasets

Sectors/Domains

# The Real Challenge

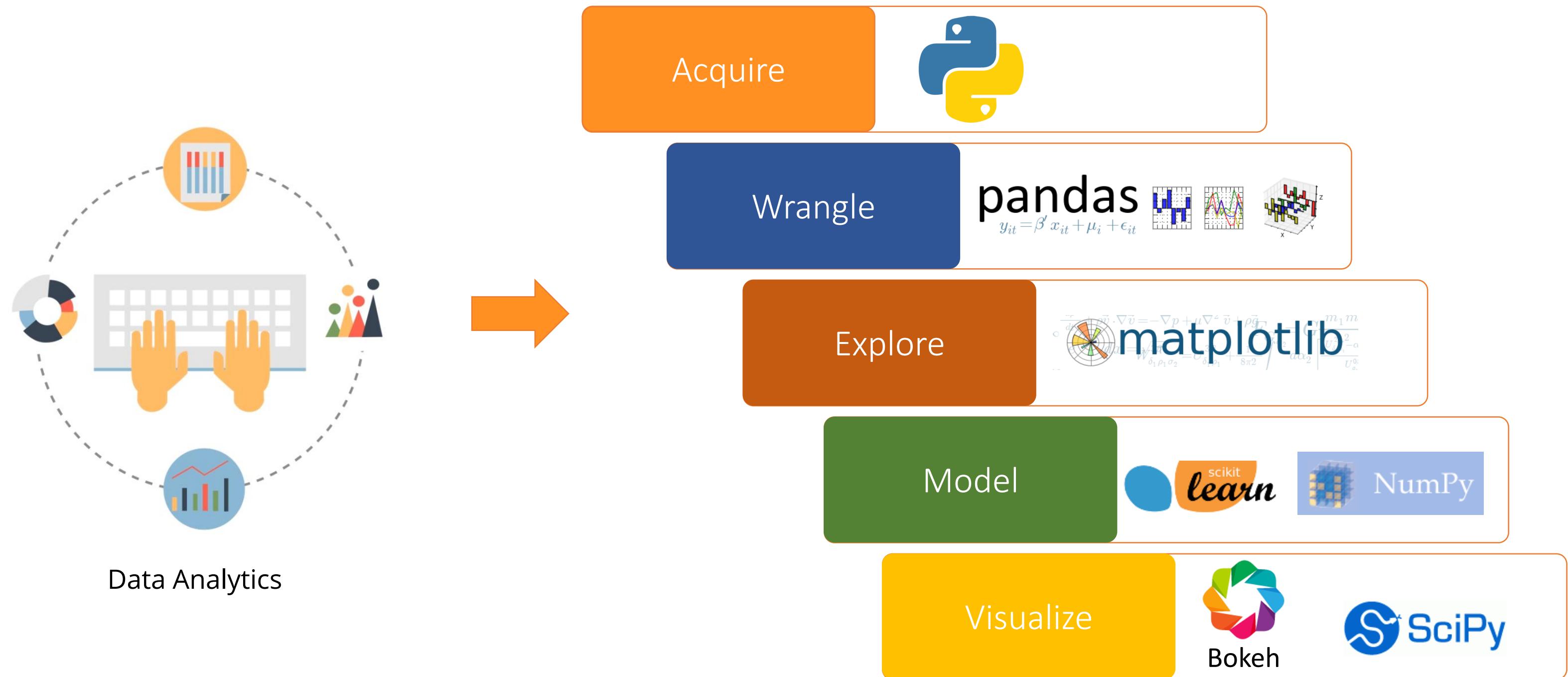
Some of the challenges Data Scientists face in the real world are listed here.

- Data quality doesn't conform to the set standards.
- Data integration is a complex task.
- Data is distributed into large clusters in HDFS, which is difficult to integrate and analyze.
- Unstructured and semi-structured data are harder to analyze.



# Data Analytics and Python

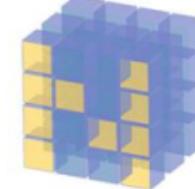
Python deals with each stage of data analytics efficiently by applying different libraries and packages.



# Python Tools and Technologies

Python is a general purpose, open source, programming language that lets you work quickly and integrate systems more effectively.

Scientific computing

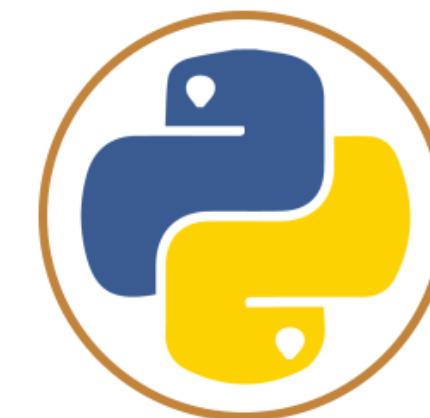


NumPy

Software for mathematics,  
science, and engineering



SciPy



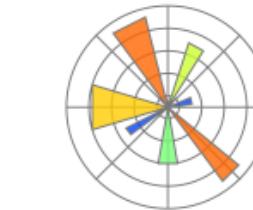
Pandas

Data analysis



Learn

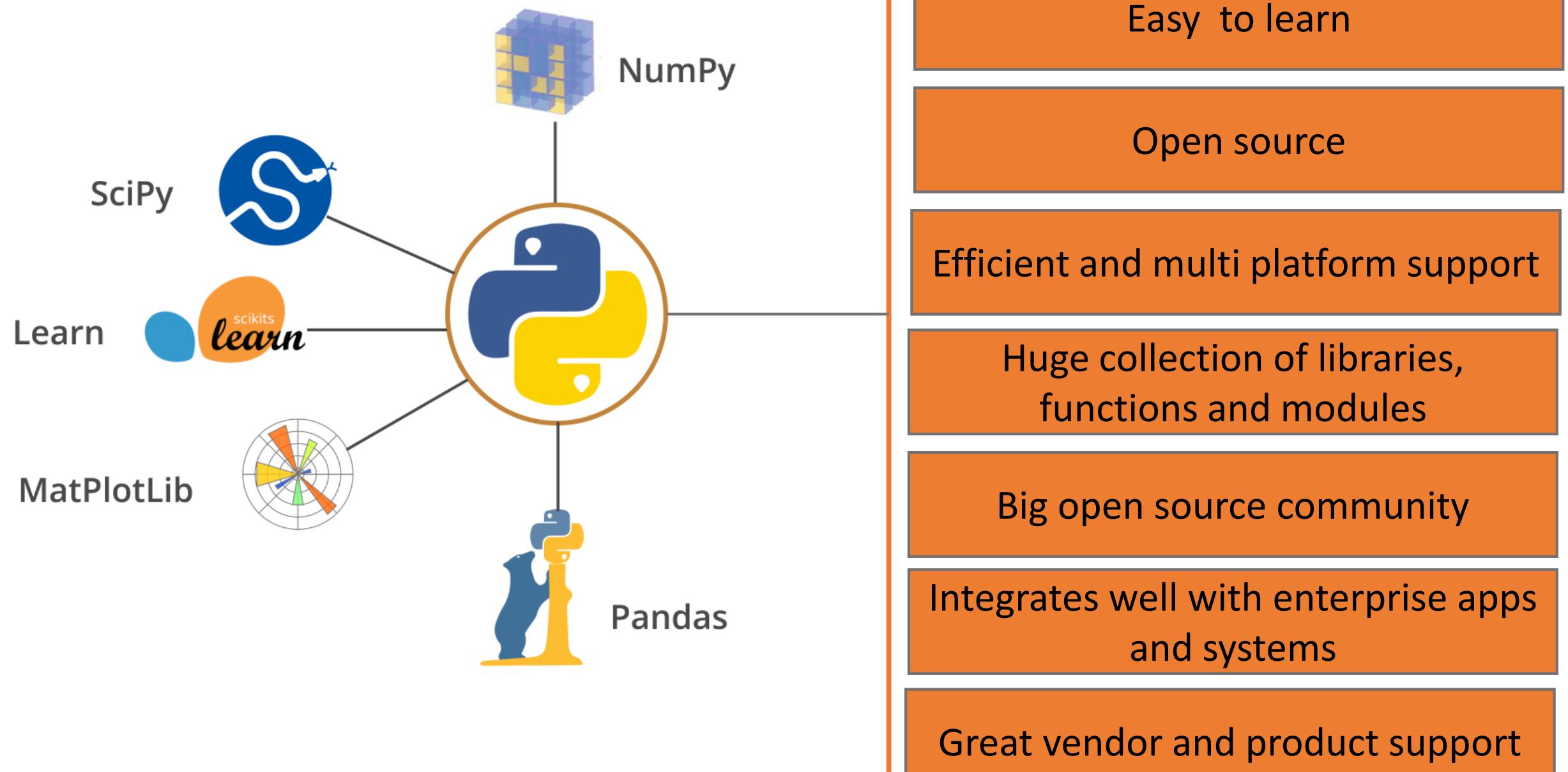
Machine learning



MatPlotLib

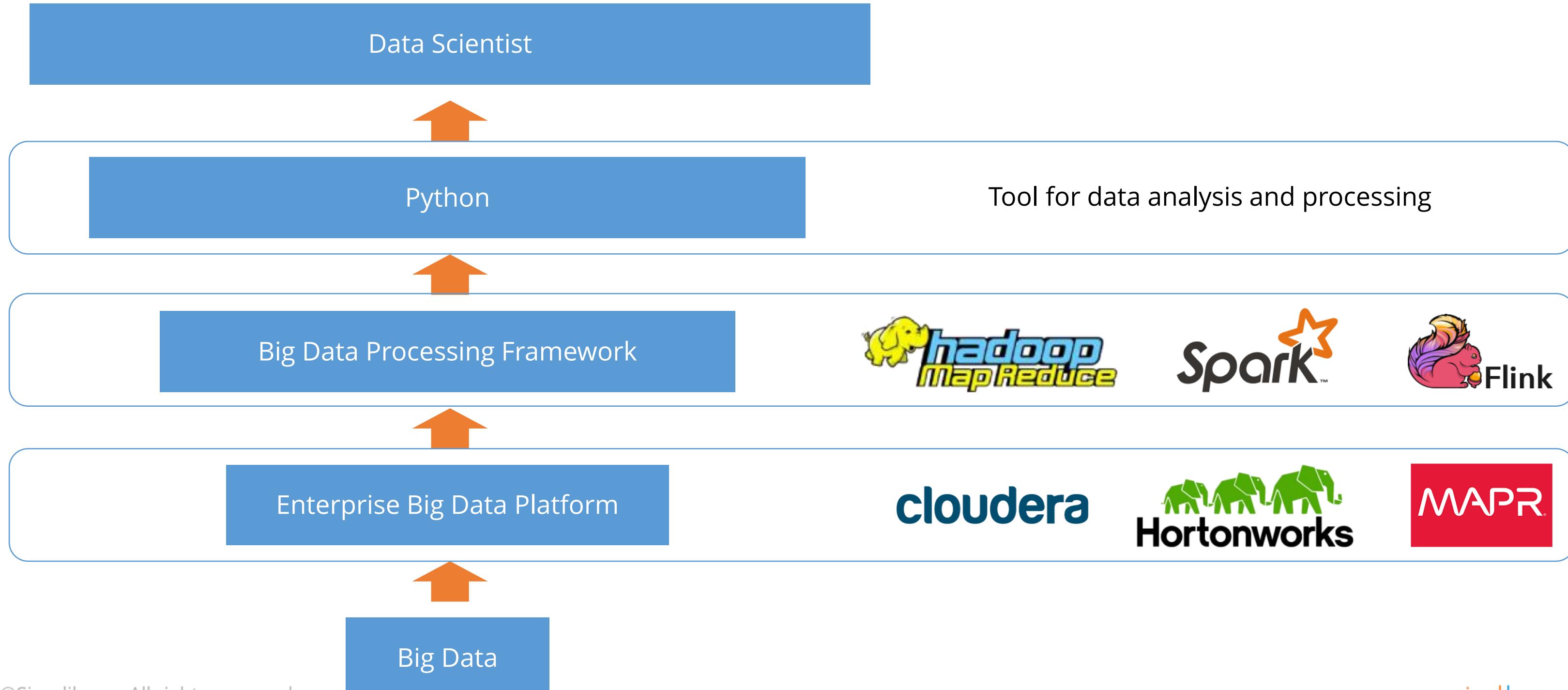
Graphics computing

# Benefits of Python



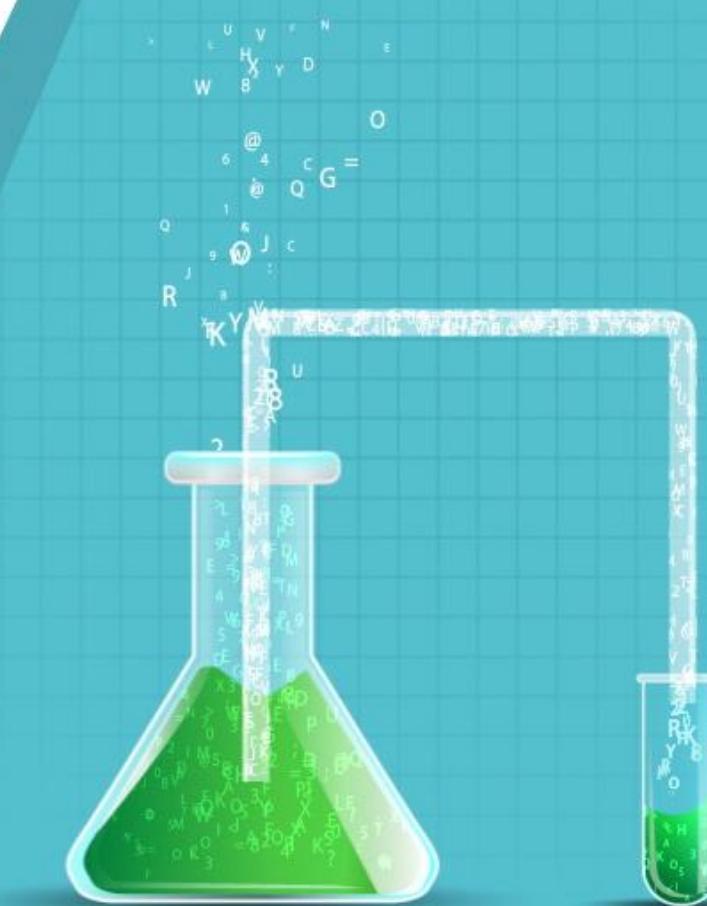
# Big Data Platforms and Processing Frameworks for Python

Python is supported by well-established data platforms and processing frameworks that help analyze data in a simple and an efficient way.



# Key Takeaways

- Data Science is a discipline that combines aspects of statistics, mathematics, programming, and domain expertise.
- Data Scientists solve big problems in public and private sectors.
- A lot of datasets are freely available to apply Data Science and turn them into data services and data products.
- Data Scientists are more in demand with the evolution of Big Data and real-time analytics.
- Python is a powerful language and a preferred tool for Data Science.





**QUIZ****1**

A Data Scientist \_\_\_\_.

- a. asks the right questions
- b. acquires data
- c. performs data wrangling and data visualization
- d. All of the above



**QUIZ****1****A Data Scientist:**

- a. Asks the right questions
- b. Acquires data
- c. Performs data wrangling and data visualization
- d. All of the above



The correct answer is **d**.

**Explanation:** A Data Scientist asks the right questions to the stakeholders, acquires data from various sources and data points, performs data wrangling that makes the data available for analysis, and creates reports and plots for data visualization.

**QUIZ**  
**2**

**The Search Engine's Autocomplete feature identifies unique and verifiable users who search for a particular keyword or phrase \_\_\_\_\_. Select all that apply.**

- a. to scrub inappropriate content.
- b. to build a Query Volume.
- c. to tag the location to a query.
- d. to find similar instances on the web.



**QUIZ  
2**

**The Search Engine's Autocomplete feature identifies unique and verifiable users who search for a particular keyword or phrase \_\_\_\_\_. Select all that apply.**

- a. to scrub inappropriate content
- b. to build a Query Volume
- c. to tag the location to a query
- d. to find similar instances on the web



The correct answer is **b, c.**

**Explanation:** The Search Engine's Autocomplete feature identifies unique and verifiable users who search for a particular keyword or phrase to build a Query Volume. It also helps identify the users' locations and tag them to the query, enabling it to be location-specific.

**QUIZ****3****What is the sequential flow of Data Analytics?**

- a. Data wrangling, exploration, modeling, acquisition, and visualization
- b. Data exploration, acquisition, modeling, wrangling, and visualization
- c. Data acquisition, wrangling, exploration, modeling, and visualization
- d. Data modeling, acquisition, exploration, wrangling, and visualization



**QUIZ****3**

## What is the sequential flow of Data Analytics?

- a. Data wrangling, exploration, modeling, acquisition, and visualization
- b. Data exploration, acquisition, modeling, wrangling, and visualization
- c. Data acquisition, wrangling, exploration, modeling, and visualization
- d. Data modeling, acquisition, exploration, wrangling, and visualization



The correct answer is **c**.

**Explanation:** In Data Analytics, the data is acquired from various sources and is then wrangled to ease its analysis. This is followed by data exploration and data modeling. The final stage is data visualization, where the data is presented and the patterns are identified.

**This concludes “Data Science Overview.”**  
The next lesson is “Data Analytics Overview.”

# DATA SCIENCE

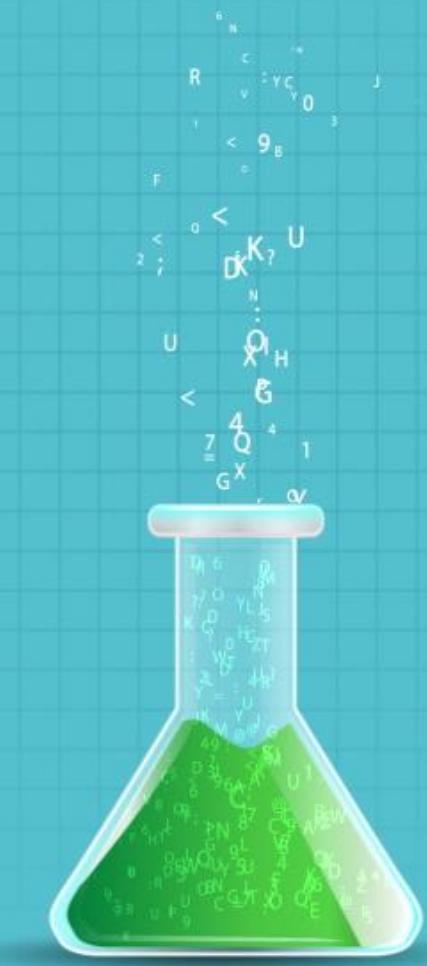
## Data Science with Python

### Lesson 2 – Data Analytics Overview



# What's In It For Me

- Data Analytics process and its steps
- Skills and tools required for Data Analysis
- Challenges of the Data Analytics Process
- Exploratory Data Analysis technique
- Data visualization techniques
- Hypothesis testing to analyze data



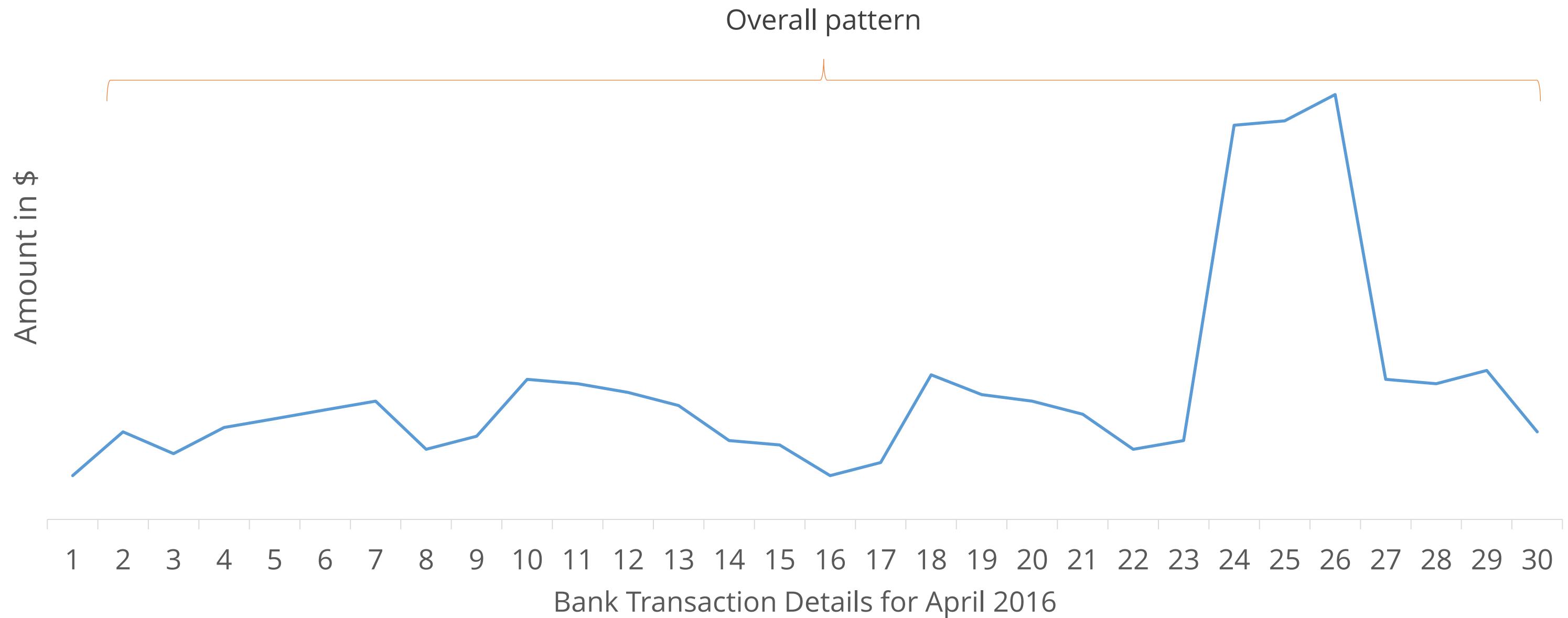
# Why Data Analytics

Data by itself is just an information source. But unless you can understand it, you will not be able to use it effectively.

Date	Description	Deposit	Withdrawal	Balance	
Apr 1	ATM Post Debit		100	\$200,000	
Apr 2	Paypal Tranfer 231054	200		\$202,000	
Apr 3	Simplilearn course fee		150	\$200,500	Information source; overall patterns not clearly visible
Apr 4	Starluck Café		210	\$198,400	
Apr 5	Walmart TX		230	\$196,100	
Apr 6	ebuy swiss watch 239		250	\$193,600	
Apr 7	Caterpillar black boots men		270	\$190,900	
Apr 8	Halo blue shirt 831		160	\$189,300	

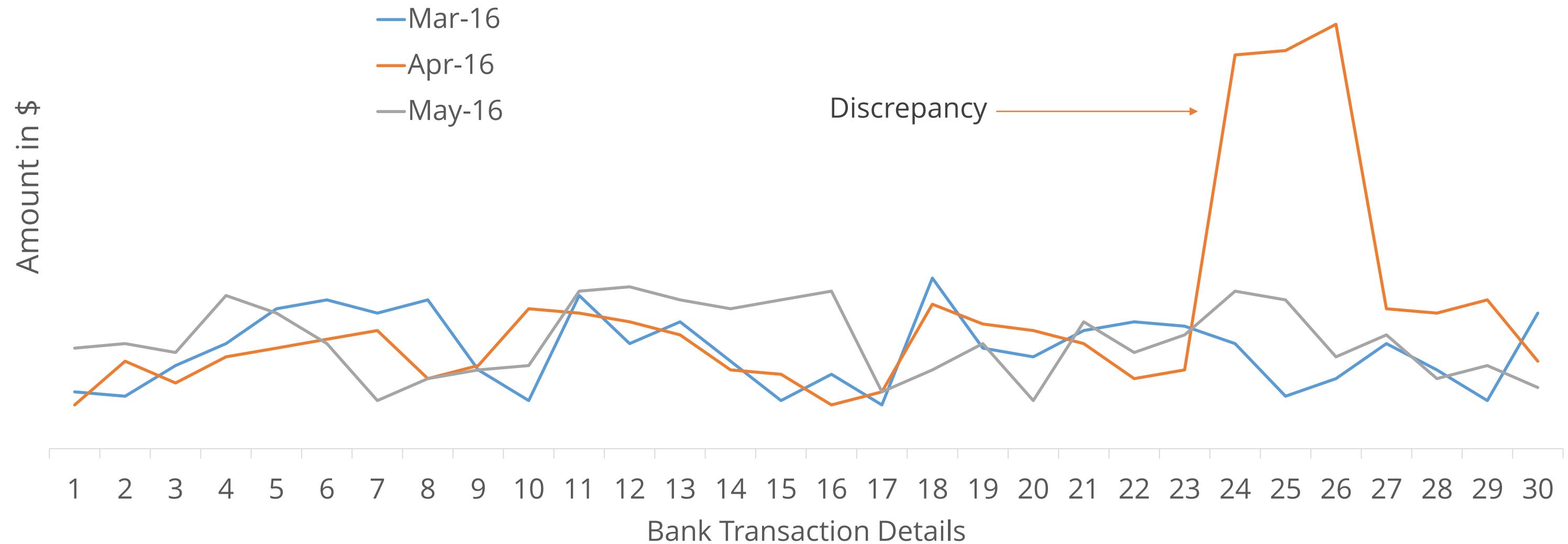
## Why Data Analytics (contd.)

When the transaction details are presented as a line chart, the deposit and withdrawal patterns become apparent.



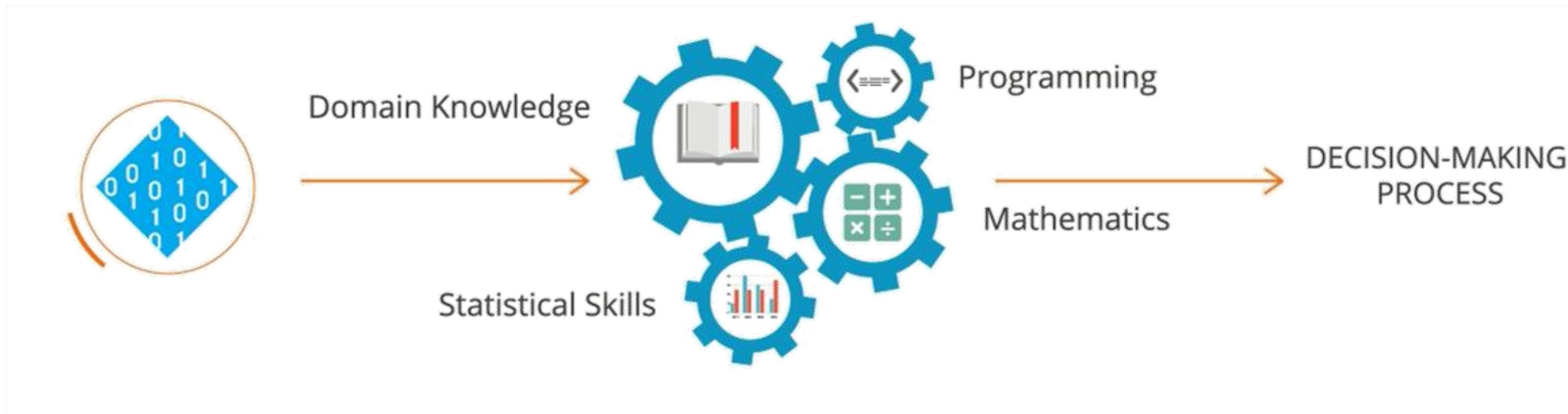
## Why Data Analytics (contd.)

When the transaction details are presented as a line chart, the deposit and withdrawal patterns become apparent. It helps view and analyze general trends and discrepancies.



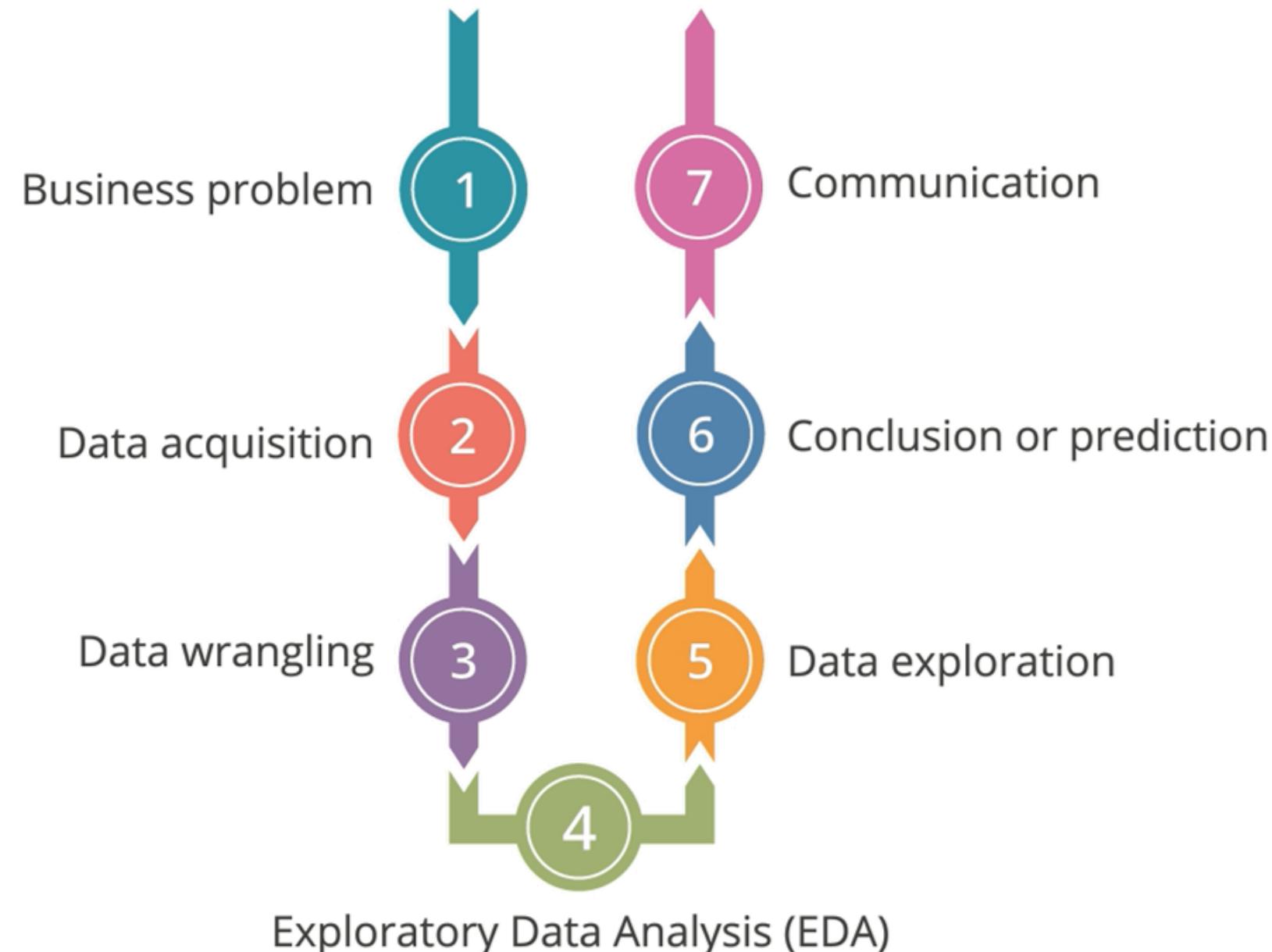
# Introduction to Data Analytics

Data Analytics is a combination of processes to extract information from datasets.



# Introduction to Data Analytics

Data Analytics is a combination of processes to extract information from datasets.



# **Business Problem**

The process of analytics begins with questions or business problems of stakeholders.



Business problems trigger the need to analyze data and find answers.

# Data Acquisition

Collect data from various sources for analysis to answer the question raised in step 1.



## Data Scientist Expertise:

- File handling
- File formats
- Web scraping



Twitter, Facebook, LinkedIn, and other social media and information sites provide streaming APIs.

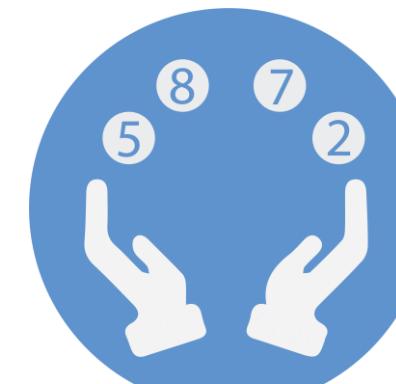
Server logs can be extracted from enterprise system servers to analyze and optimize application performance.

# Data Wrangling and Exploration

Data wrangling is the most important phase of the data analytic process.



Data cleansing



Data manipulation



Data discovery



Data pattern



Data Wrangling



Data Exploration

# Data Wrangling—Challenges

This phase includes data cleansing, data manipulation, data aggregation, data split, and reshaping of data.



Causes of challenges in the data wrangling phase:

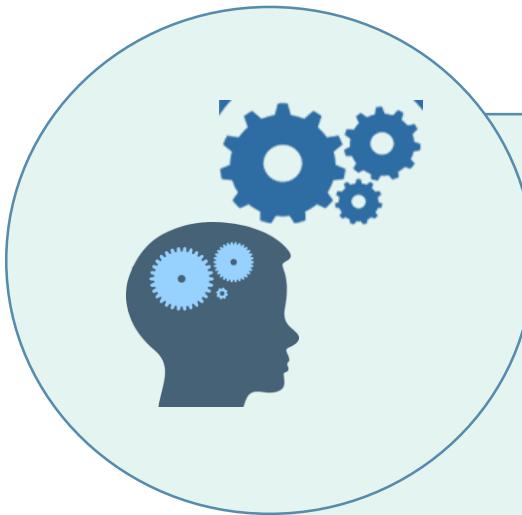
- Unexpected data format
- Erroneous data
- Voluminous data to be manipulated
- Classifying data into linear or clustered
- Determining relationship between observation, feature, and response



Data wrangling is the most challenging phase and takes up 70% of the data scientist's time.

# Data Exploration—Model Selection

This phase includes data cleansing, data manipulation, data aggregation, data split, and reshaping of data.



## Model selection

- Based on the overall data analysis process
- Should be accurate to avoid iterations
- Depends on pattern identification and algorithms
- Depends on hypothesis building and testing
- Leads to building mathematical statistical functions

# Exploratory Data Analysis (EDA)

Let's take a look at the exploratory data analysis phase.



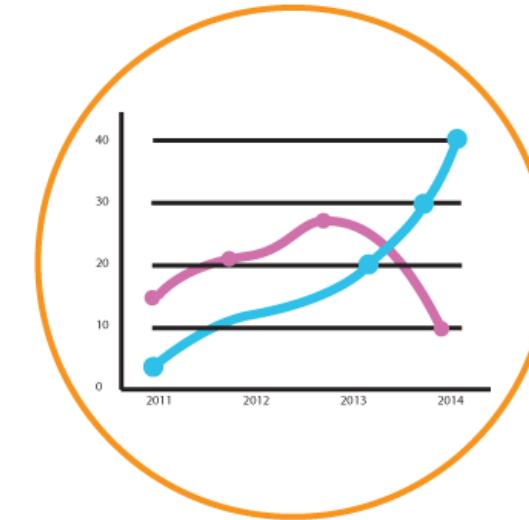
## APPROACH

EDA approach studies the data to recommend suitable models that best fit the data.



## FOCUS

The focus is on data; its structure, outliers, and models suggested by the data.



## ASSUMPTIONS

EDA techniques make minimal or no assumptions. They present and show all the underlying data without any data loss.



## EDA TECHNIQUES

**Quantitative:** Provides numeric outputs for the inputted data  
**Graphical:** Uses statistical functions for graphical output

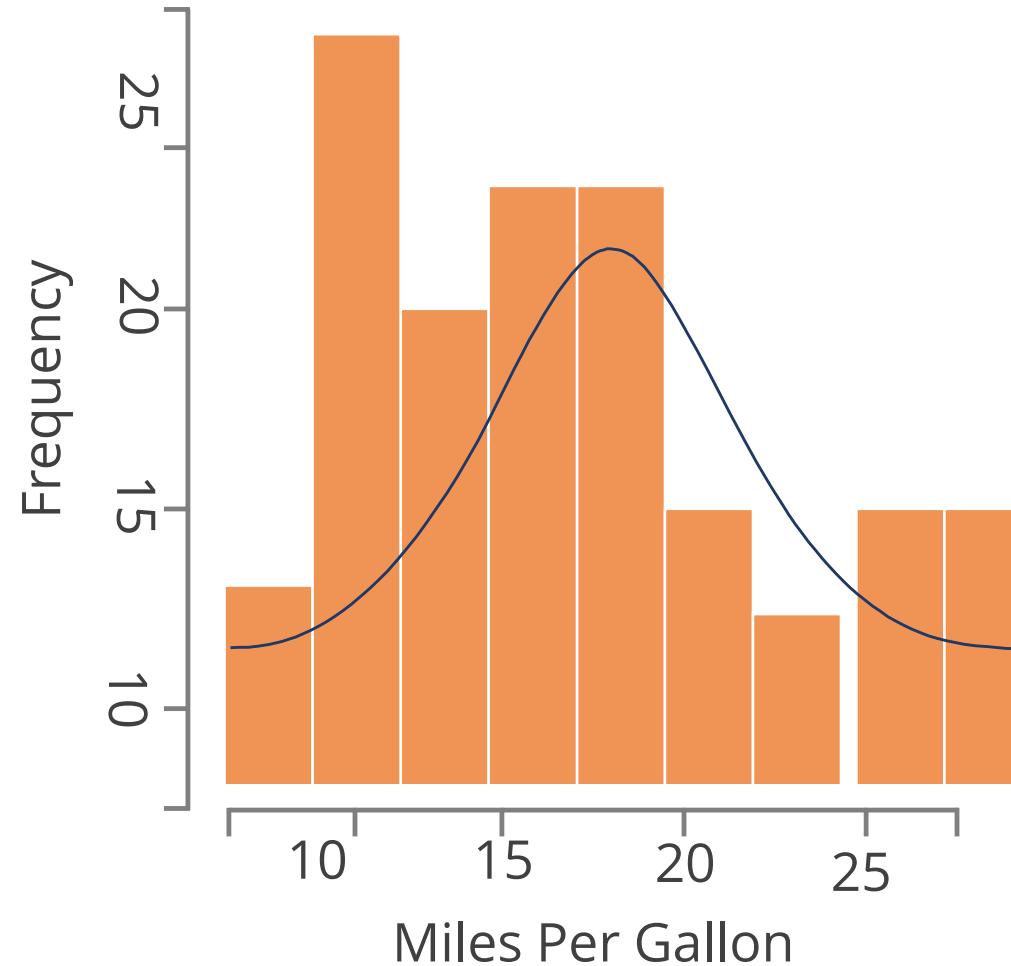
# EDA— Quantitative Technique

EDA – Quantitative technique has two goals, measurement of central tendency and spread of data.

Measurement of Central Tendency	
Mean	Mean is the point which indicates how centralized the data points are. <ul style="list-style-type: none"><li>• Suitable for symmetric distributions</li></ul>
Median	Median is the exact middle value. <ul style="list-style-type: none"><li>• Suitable for skewed distributions and for catching outliers in the dataset</li></ul>
Mode	Mode is the most common value in the data (frequency).
Measurement of Spread	
Variance	Variance is approximately the mean of the squares of the deviations.
Standard Deviation	Standard deviation is the square root of the variance.
Inter-quartile Range	Inter-quartile range is the distance between the 75 <sup>th</sup> and 25 <sup>th</sup> percentile. It's essentially the middle 50% of the data.

## EDA – Graphical Technique

Histograms and Scatter Plots are two popular graphical techniques to depict data.



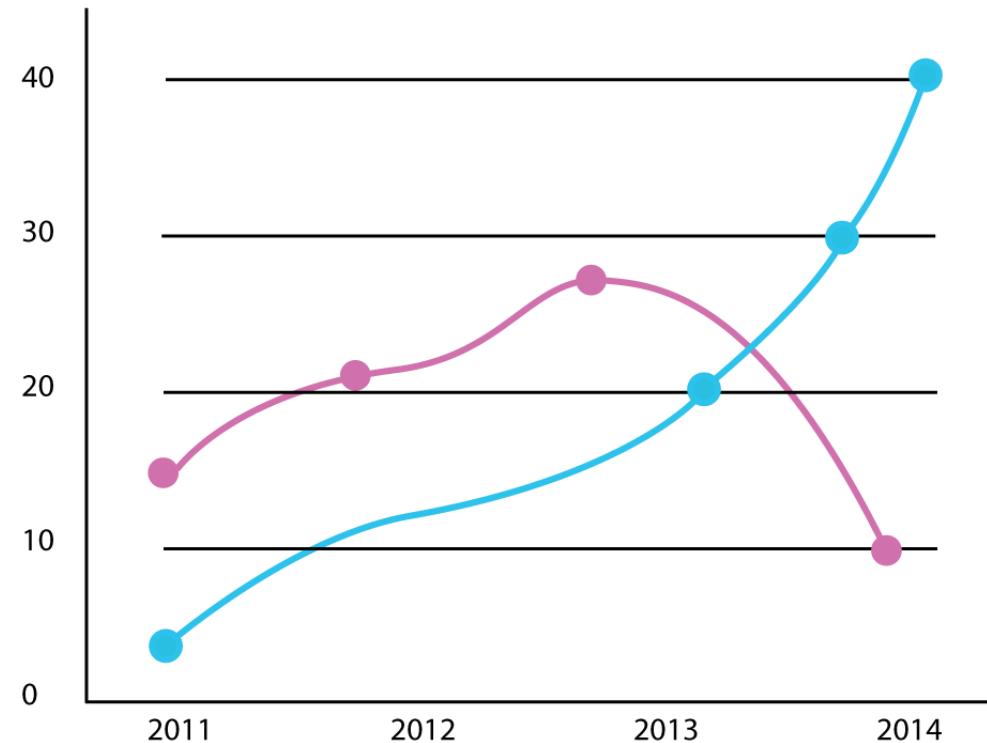
Histogram graphically summarizes the distribution of a univariate dataset.

It shows:

- the center or location of data (mean, median, or mode)
- the spread of data
- the skewness of data
- the presence of outliers
- the presence of multiple modes in the data

# EDA – Graphical Technique

Histograms and Scatter Plots are two popular graphical techniques to depict data.



A Scatter plot represents relationships between two variables. It can answer these questions visually:

- Are variables X and Y related?
- Are variables X and Y linearly related?
- Are variables X and Y non-linearly related?
- Does change in variation of Y depend on X?
- Are there outliers?



# Knowledge Check

KNOWLEDGE  
CHECK

What is the goal of data acquisition?

*Select all that apply.*

- a. Collect data from various data sources
- b. Answer business questions through graphics
- c. Collect web server logs
- d. Scrape the web through web APIs



KNOWLEDGE  
CHECK

What is the goal of data acquisition?

*Select all that apply.*

- a. Collect data from various data sources
- b. Answer business questions through graphics
- c. Collect web server logs
- d. Scrape web through web APIs



The correct answer is **a, c ,d**

**Explanation:** Data acquisition is a process to collect data from various data sources such as RDBMS, NoSQL databases, web server logs and also scrape the web through web APIs.

KNOWLEDGE  
CHECK

What is the Exploratory Data Analysis technique?

*Select all that apply.*

- a. Analysis of data using quantitative techniques
- b. Conducted only on a small subset of data
- c. Analysis of data using graphical techniques
- d. Suggests admissible models that best fit the data



KNOWLEDGE  
CHECK

What is the Exploratory Data Analysis technique?

*Select all that apply.*

- a. Analysis of data using quantitative techniques
- b. Conducted only on small subset of data
- c. Analysis of data using graphical techniques
- d. Suggests models that best fit the data



The correct answer is **a, c, d**.

**Explanation:** Most EDA techniques are graphical in nature with a few quantitative techniques and also suggest models that best fit the data. They use almost the entire data with minimum and no assumptions.

## Conclusion or Predictions

This step involves reaching a conclusion and making predictions based on the data analysis.

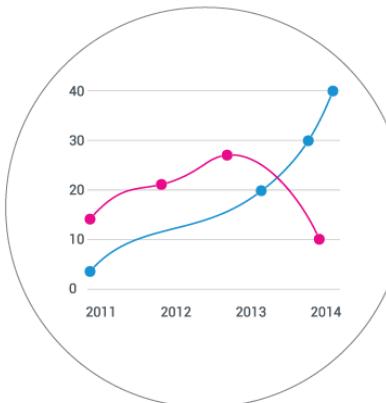


This phase:

- Involves heavy use of mathematical and statistical functions
- Requires model selection, training, and testing to help in forecasting
- Is called “machine learning” as data analysis is fully or semi-automated with minimal or no human intervention

# Meaning of Hypothesis

Hypothesis is used to establish the relationship between dependent and independent variables.



Data Exploration Stage

Hypothesis building begins in the data exploration stage but becomes more mature in the conclusion or prediction phase.



Conclusion and Prediction

## Key Considerations of Hypothesis Building

- Testable explanations of a problem or observation
- Used in quantitative and qualitative analyses to provide research solutions
- Involves two variables, one dependent on another
- Independent variable manipulated by the researcher
- Dependent variable changes when the independent variable changes

# Hypothesis Building Using Feature Engineering

Domain knowledge leads to hypothesis building using feature engineering.



Feature engineering involves domain expertise to:

- Make sense of data
- Construct new features from raw data automatically
- Construct new features from raw data manually

# Hypothesis Building Using a Model

There are three phases to hypothesis building which are model building, model evaluation, and model deployment.

## Phase 1: Model Building

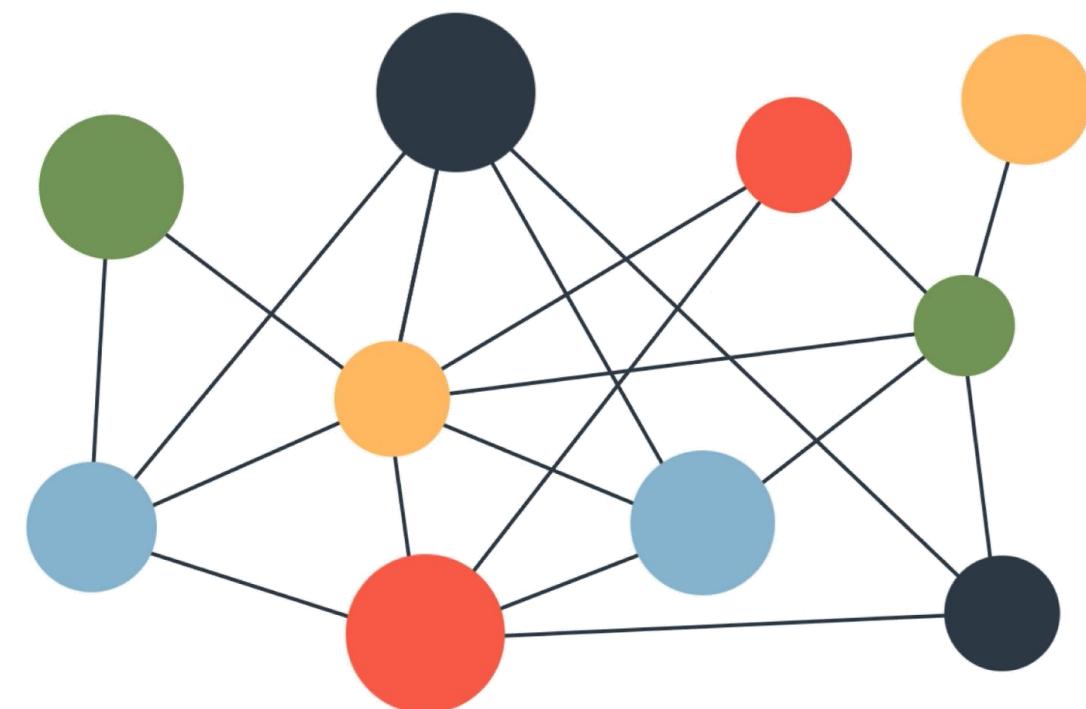
- Identify best input variables
- Evaluate the model's capacity to forecast with these variables

## Phase 2: Model Evaluation

- Train and test the model for accuracy
- Optimize model accuracy, performance, and comparisons with other models

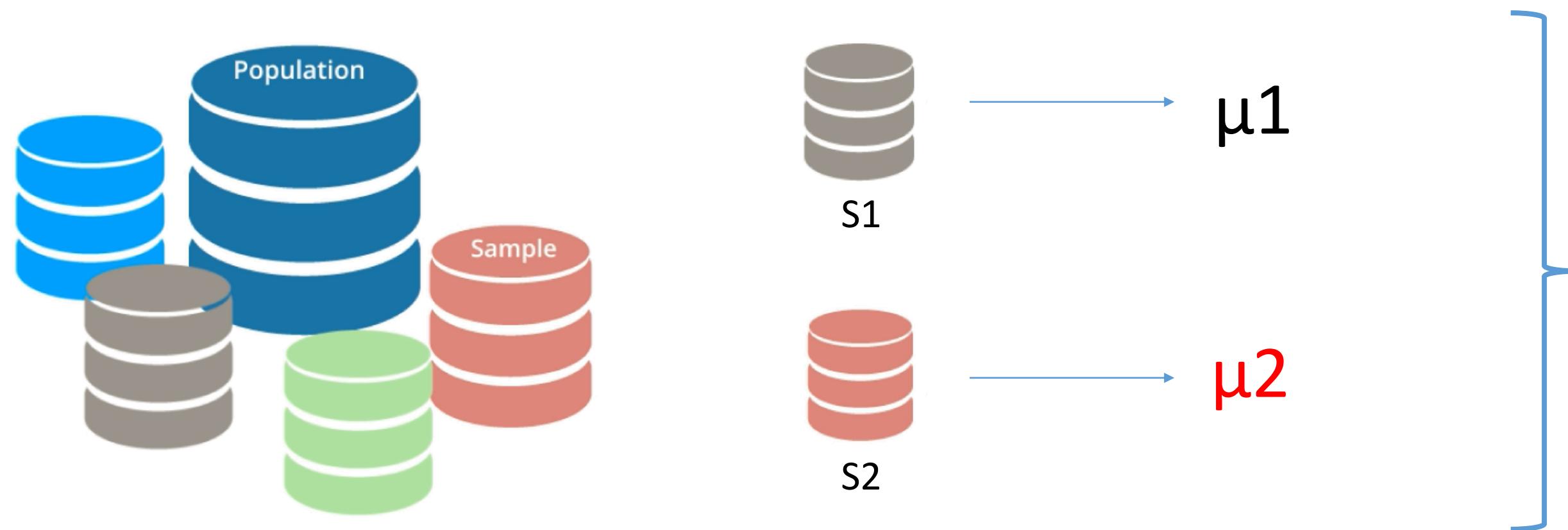
## Phase 3: Model Deployment

- Use the model for prediction
- Use the model to compare actual outcome with expectations



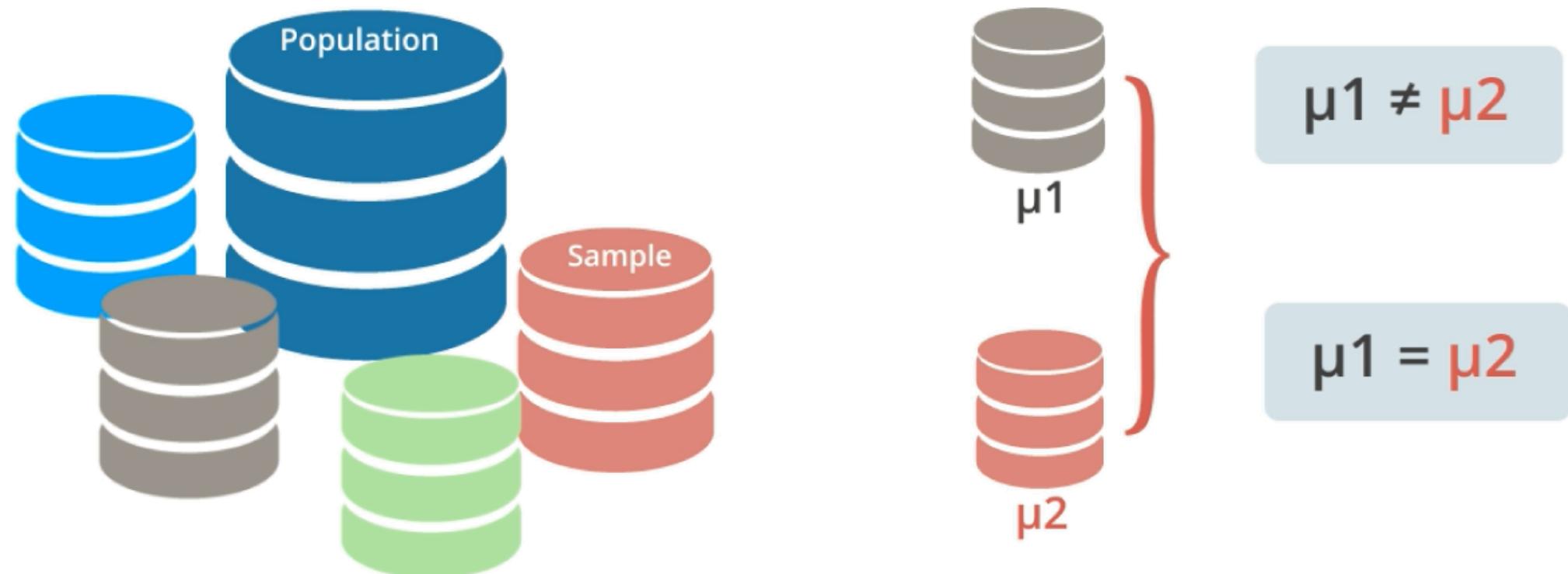
# Hypothesis Testing

Draw two samples from the population and calculate the difference between their means.



# Hypothesis Testing

Draw two samples from the population and calculate the difference between their means.



## Alternative Hypothesis

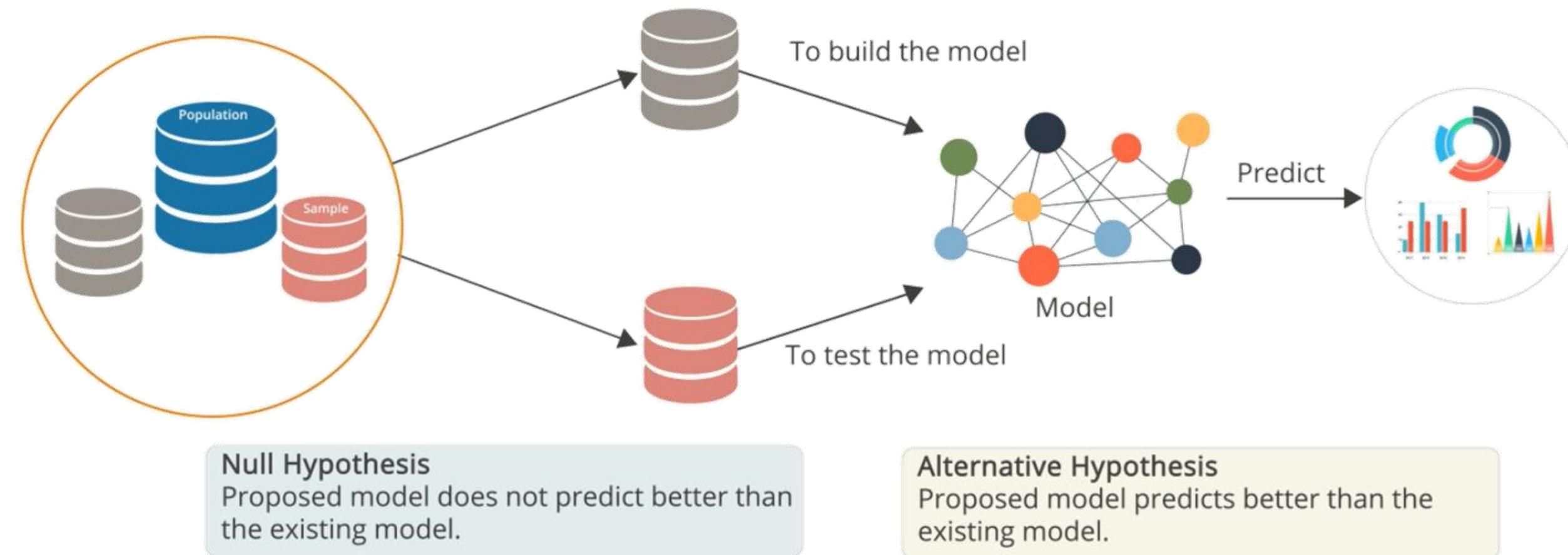
- Proposed model outcome is accurate and matches the data.
- There is a difference between the means of S1 and S2.

## Null Hypothesis

- Opposite of the alternative hypothesis.
- There is no difference between the means of S1 and S2.

# Hypothesis Testing Process

Choosing the training and test dataset and evaluating them with the null and alternative hypothesis.



Usually the training dataset is between 60% and 80% of the big dataset and the test dataset is between 20% and 40% of the big dataset.

# Communication

Data analysis process and results are presented to stakeholders.

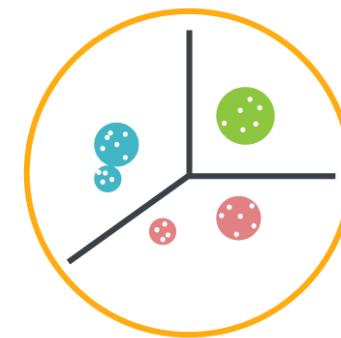


Forms of Data analysis presentation:

- Visual graphs
- Plotting maps
- Reports
- Whitepaper reports
- PowerPoint presentations

# Data Visualization

Data visualization techniques are used for effective communication of data.



## Benefits of data visualization:

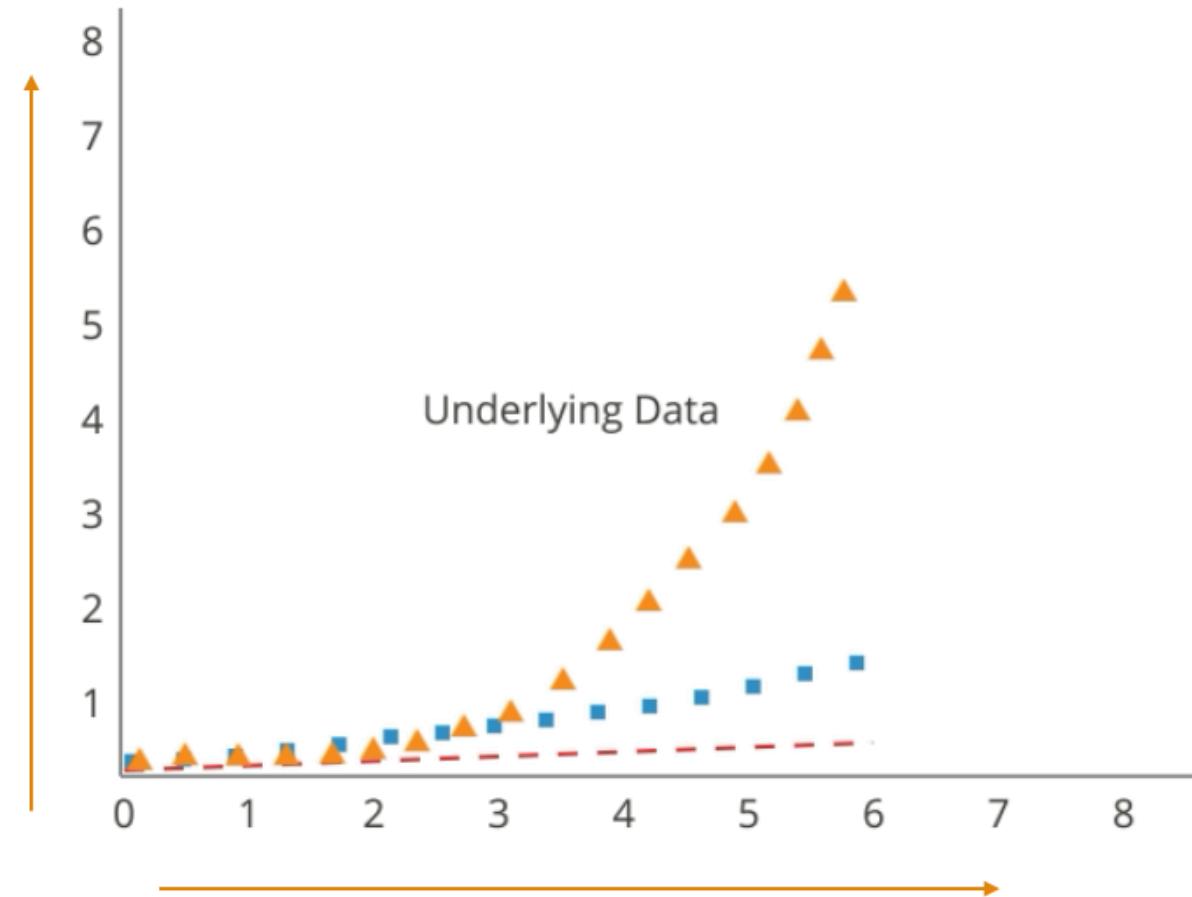
- Simplifies quantitative information through visuals
- Shows the relationship between data points and variables
- Identifies patterns
- Establishes trends

## Examples of data visualization:

- Presenting information about new and existing customers on the website and their behavior when they access the website
- Representing web traffic pattern for the website, for example, more activity on the website in the morning than in the evening

# Plotting

Plotting is a data visualization technique used to represent underlying data through graphics.

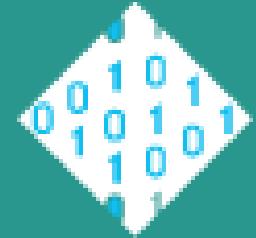


Features of plotting:

- Plotting is like telling a story about data using different colors, shapes, and sizes.
- Plotting shows the relationship between variables.
- Example:
  - Change in value of Y results in change in value of X.
  - X is independent of y.

# Data Types for Plotting

There are various data types used for plotting.



Numerical Data

There are two types of numerical data:  
Discrete Data – Distinct or counted values  
Example: Number of employees in a company or number of students in a class  
Continuous Data – Values within a range that can be measured  
Example: Height can be measured in feet or inches and weight can be measured in pounds or kilograms



Categorical Data

There are two types of categorical data:  
Cluster or group – Grouped values  
Example: Students can be divided into different groups based on height – Tall, Medium, and Short  
Ordinal data – Grouped values as per ranks  
Example: A ranking system; a five-point scale with ranks like “Agree,” “Strongly agree,” and “Disagree”

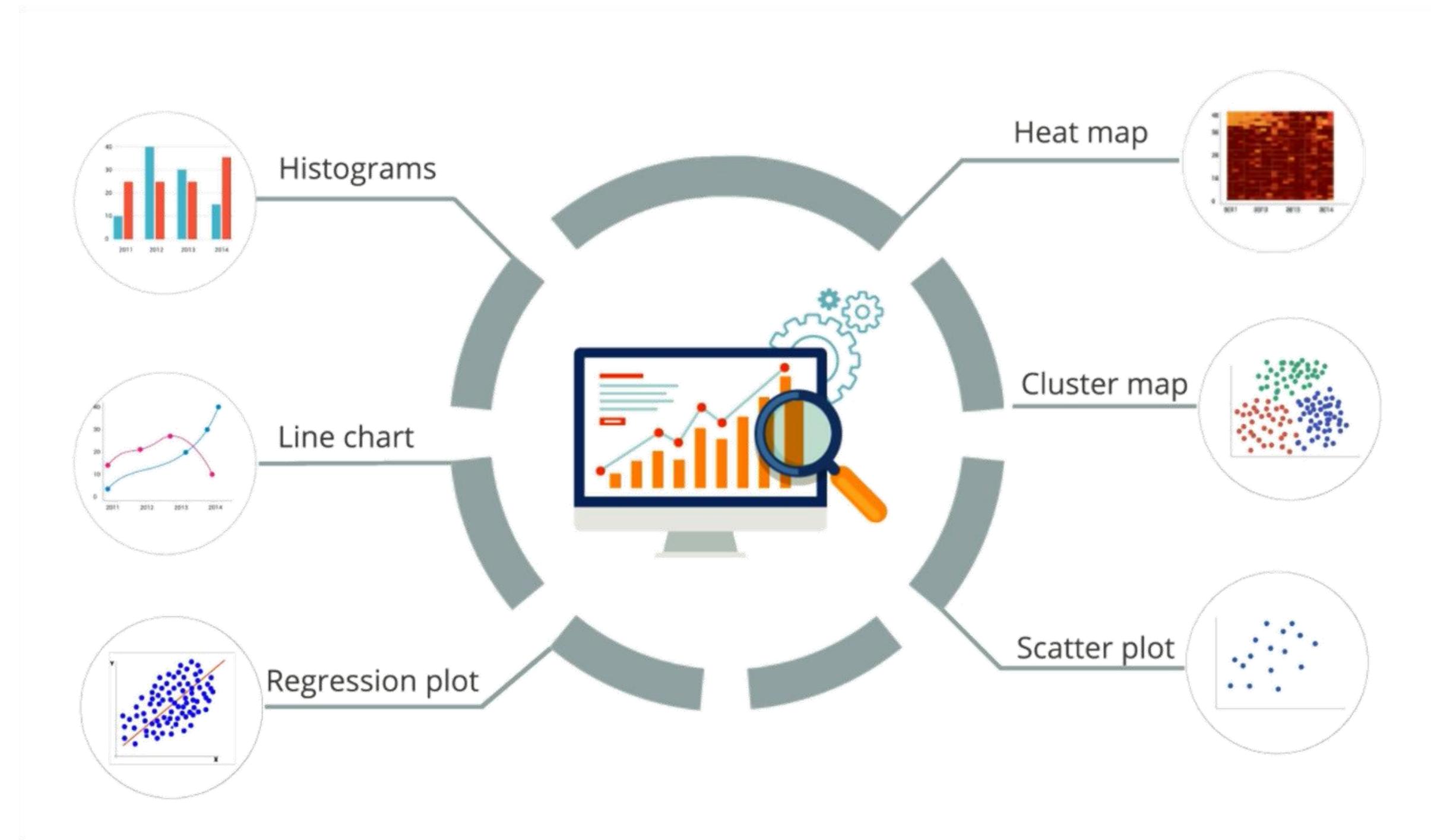


Time Series

Data measured in time blocks such date, month, year, and time (hours, minutes, and seconds)

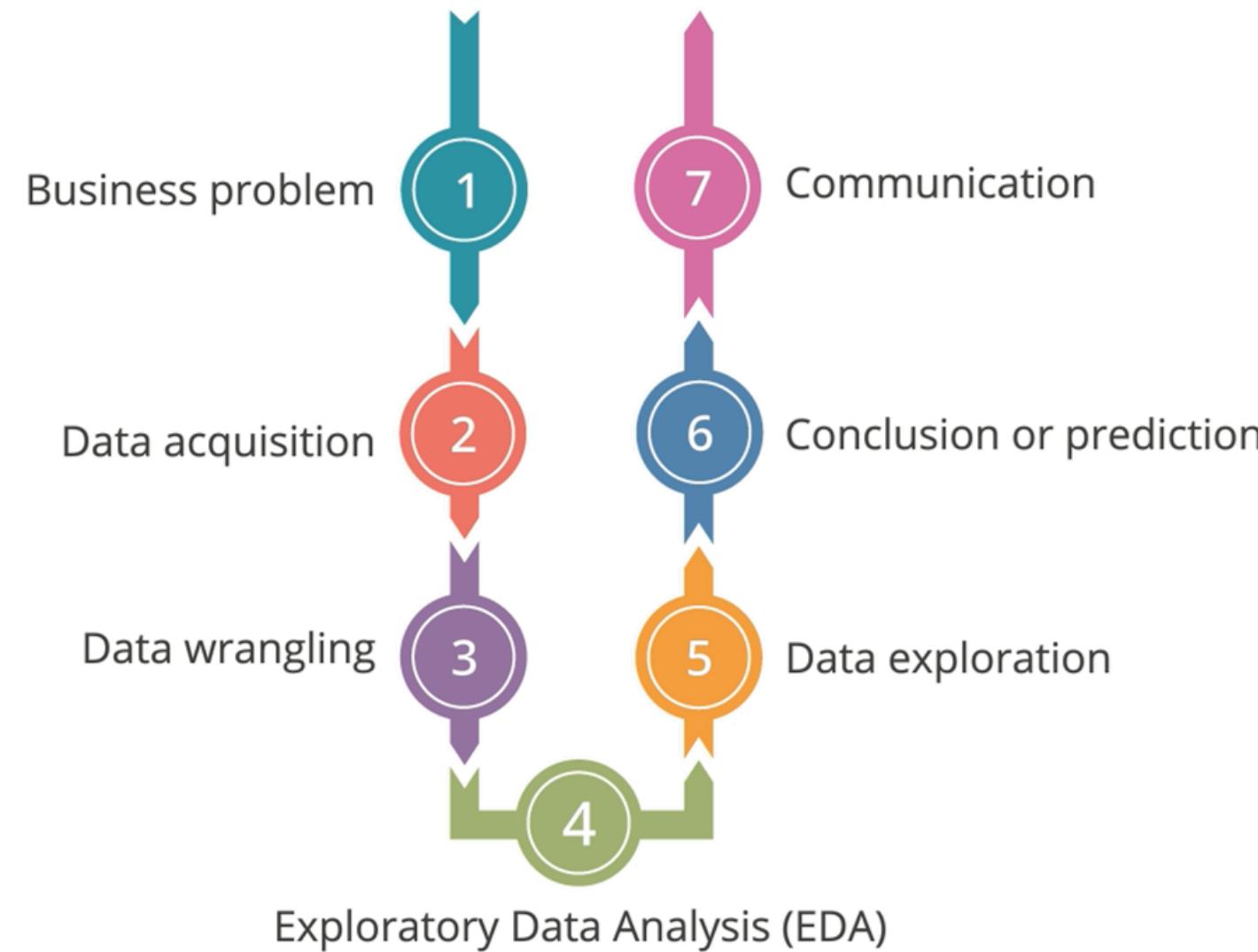
# Types of Plot

Different data types can be visualized using various plotting techniques.



# Data Analytics – An Iterative Process

Data analytics is an iterative process involving tracing back the steps, often to ensure that you are on the right track.



Process Result: Question is answered or business problem is solved.

# Data Analytics – Skills and Tools

Skills and tools required for each step of the data analysis process.

	<b>Question or Business Problem</b>		<b>Data Acquisition</b>		<b>Data Wrangling</b>		<b>Data Exploration</b>		<b>Conclusion or Predictions</b>		<b>Communication or Data Visualization</b>
<ul style="list-style-type: none"><li>Ability to ask appropriate questions and know the business</li><li>Domain knowledge</li><li>Passion for data</li><li>Analytical approach</li></ul>		<ul style="list-style-type: none"><li>BeautifulSoup for web scraping</li><li>CSV or other file knowledge</li><li>NumPy</li><li>Pandas</li><li>Database</li></ul>	<ul style="list-style-type: none"><li>CSV or other file knowledge</li><li>NumPy</li><li>Pandas</li><li>Database</li><li>SciPy</li></ul>		<ul style="list-style-type: none"><li>NumPy</li><li>SciPy</li><li>Pandas</li><li>Matplotlib</li></ul>		<ul style="list-style-type: none"><li>Scikit-Learn – the main machine learning library</li><li>CSV or other file knowledge</li><li>NumPy</li><li>Pandas</li><li>Database</li><li>SciPy</li></ul>		<ul style="list-style-type: none"><li>Pandas</li><li>Database</li><li>Matplotlib</li><li>PPT</li><li>CSV or other file knowledge</li></ul>		



# Knowledge Check

KNOWLEDGE  
CHECK

Which plotting technique is used for continuous data?

*Select all that apply.*

- a. Regression plot
- b. Line chart
- c. Histogram
- d. Heat map



KNOWLEDGE  
CHECK

Which plotting technique is used for continuous data?

*Select all that apply.*

- a. Regression plot
- b. Line chart
- c. Histogram
- d. Heat map



The correct answer is **b , c** .

**Explanation:** Line charts and histograms are used to plot continuous data.



QUIZ  
1

Which Python library is the main machine learning library?

- a. Pandas
- b. Matplotlib
- c. Scikit-learn
- d. NumPy



QUIZ  
1

Which Python library is the main machine learning library?

- a. Pandas
- b. Matplotlib
- c. Scikit-learn
- d. NumPy



The correct answer is **c**.

**Explanation:** SciKit-learn is the main machine library in Python.

QUIZ  
2

Which of the following includes data transformation, merging, aggregation, group by operation, and reshaping?

- a. Data Acquisition
- b. Data Visualization
- c. Data Wrangling
- d. Machine learning



QUIZ  
2

Which of the following includes data transformation, merging, aggregation, group by operation, and reshaping?

- a. Data Acquisition
- b. Data Visualization
- c. Data Wrangling
- d. Machine learning



The correct answer is **C**.

**Explanation:** Data wrangling includes data transformation, merging, aggregation, group by operation, and reshaping.

QUIZ  
3

Which measure of central tendency is used to catch outliers in the data?

- a. Mean
- b. Median
- c. Mode
- d. Variance



QUIZ  
3

Which measure of central tendency is used to catch outliers in the data?

- a. Mean
- b. Median
- c. Mode
- d. Variance



The correct answer is **b**.

**Explanation:** Median is the exact middle value and most suitable to catch outliers.

QUIZ  
4

In hypothesis testing, the proposed model is built on:

- a. the entire dataset.
- b. the test dataset.
- c. a small subset.
- d. the training dataset.



QUIZ  
4

In hypothesis testing, the proposed model is built on:

- a. the entire dataset.
- b. the test dataset.
- c. a small subset.
- d. the training dataset.



The correct answer is **d**.

**Explanation:** The proposed model is built on the training dataset in hypothesis testing.

QUIZ  
5

Beautiful soup library is used for \_\_\_\_.

- a. data wrangling
- b. web scraping
- c. plotting
- d. machine learning



QUIZ  
5

Beautiful soup library is used for \_\_\_\_.

- a. data wrangling
- b. web scraping
- c. plotting
- d. machine learning.

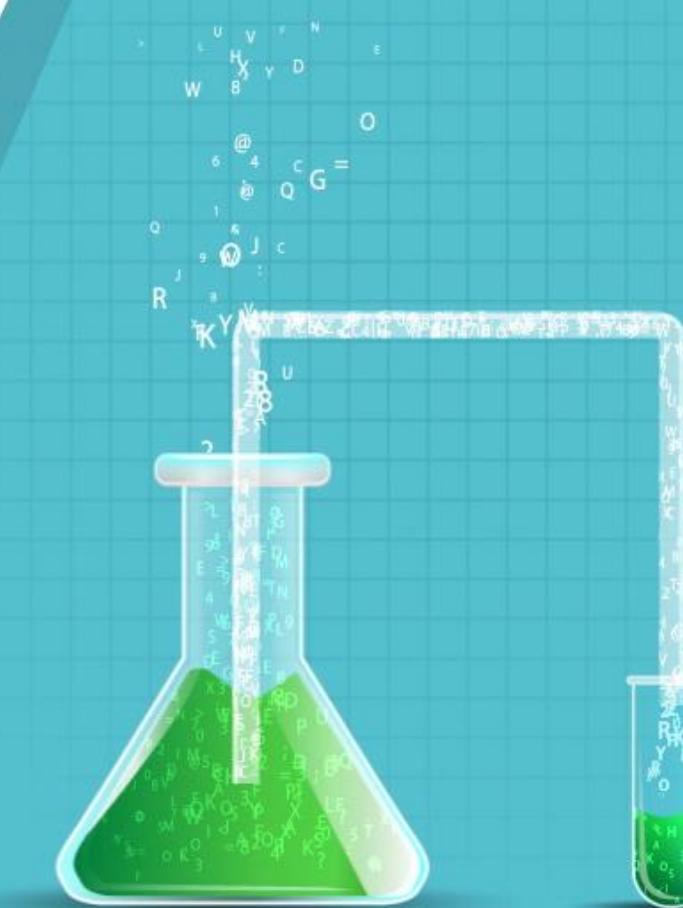


The correct answer is **b**.

**Explanation:** BeautifulSoup is used for web scraping and mainly used in the data acquisition phase.

# Key Takeaways

- Data analytics is used to solve business problems.
- Data analysis requires a number of skills and tools.
- Data wrangling, data exploration, and model selection processes are challenging.
- EDA includes quantitative and graphical techniques.
- Data visualization helps show data characteristics and patterns effectively.
- Hypothesis testing establishes the relationship between dependent and independent variables in data analytics.



**This concludes “Data Analytics”**

The next lesson is “Statistical Analysis and Business Applications”

DATA  
SCIENCE

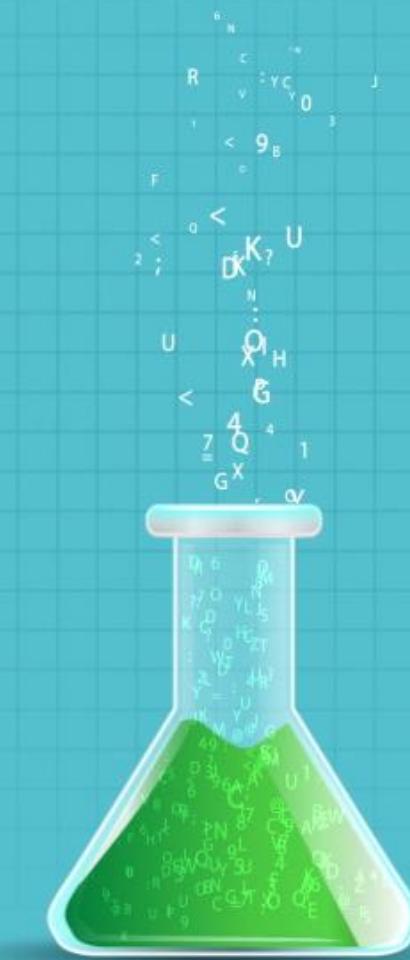
# Data Science with Python

## Lesson 3 – Statistical Analysis and Business Applications



# What You'll Learn

- The difference between statistical and non-statistical analysis
- The two major categories of statistical analysis and their differences
- The statistical analysis process
- Mean, median, mode, and percentile
- Data distribution and the various methods of representing it
- Hypothesis testing and the Chi square test
- Types of frequencies
- Correlation matrix and its uses



# Introduction to Statistics

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data.

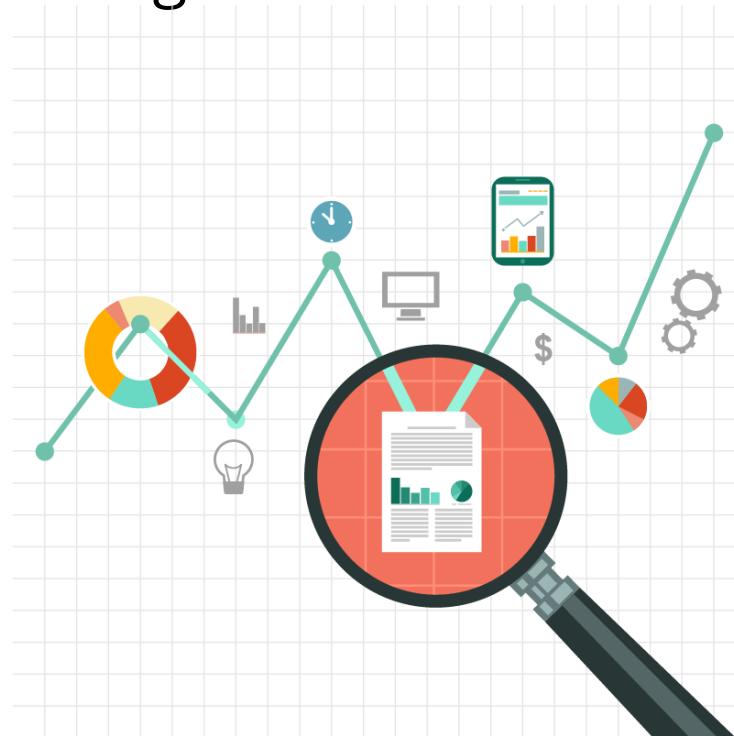


# Introduction to Statistics

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data.

Tools available to analyze data:

- Statistical principles
- Functions
- Algorithms



What you can do using statistical tools:

- Analyze the primary data
- Build a statistical model
- Predict the future outcome

# Statistical and Non-statistical Analysis

## Statistical Analysis



Statistical Analysis is:

- scientific
- based on numbers or statistical values
- useful in providing complete insight to the data

## Non-statistical Analysis



Non-statistical Analysis is:

- based on very generic information
- exclusive of statistical or quantitative analysis

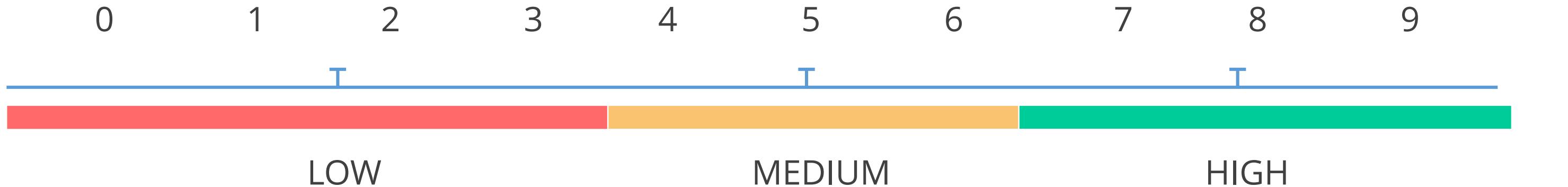
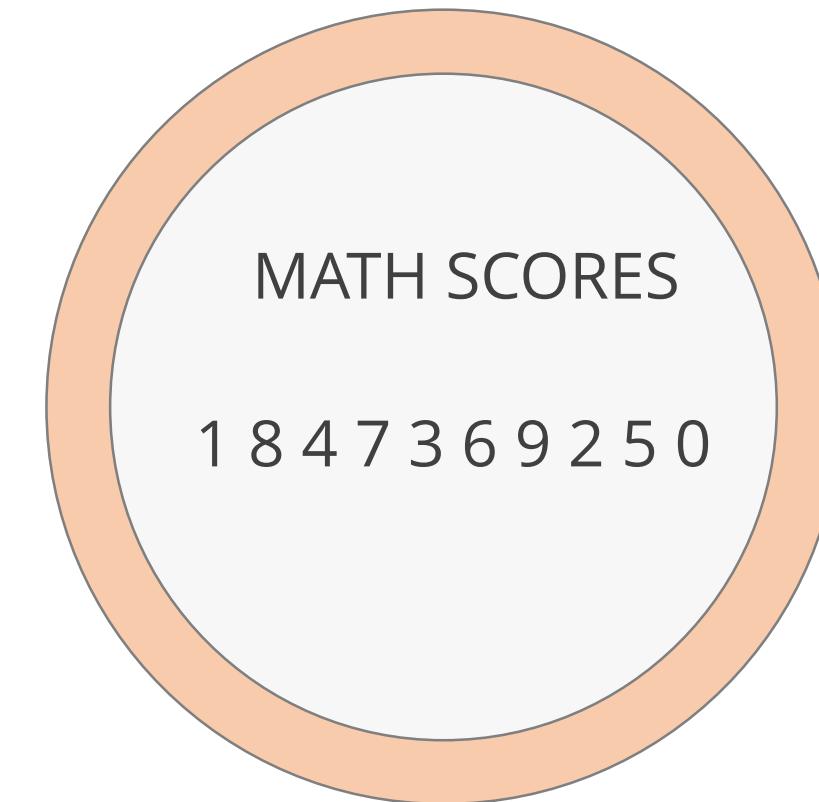


Although both forms of analysis provide results, quantitative analysis provides more insight and a clearer picture. This is why statistical analysis is important for businesses.

# Major Categories of Statistics

There are two major categories of statistics: Descriptive analytics and inferential analytics

Descriptive analysis organizes the data and focuses on the main characteristics of the data.



# Major Categories of Statistics

Inferential analytics uses the probability theory to arrive at a conclusion.



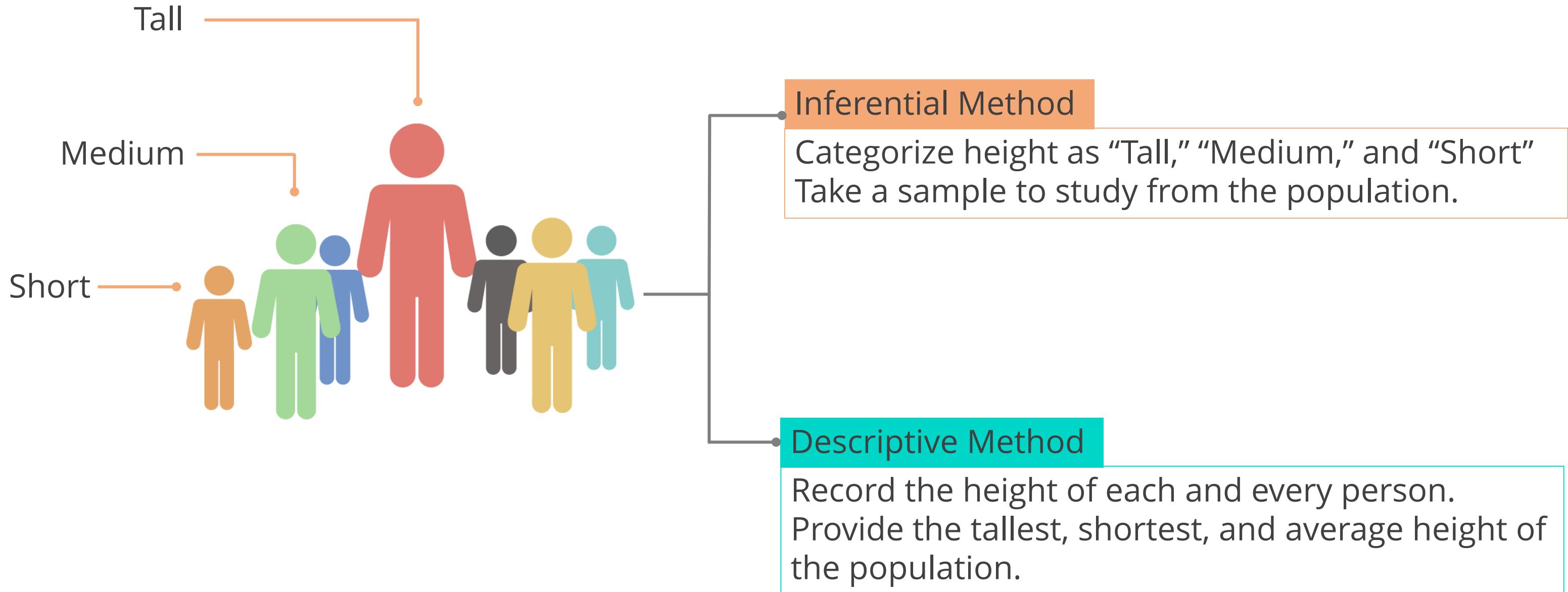
- Random sample is drawn from the population
- Used to describe and make inferences about the population



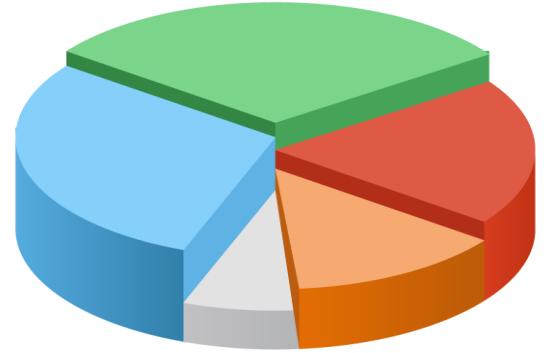
Inferential analytics is valuable when it is not possible to examine each member of the population.

# Major Categories of Statistics – An Example

Study of the height of the population



# Statistical Analysis Considerations



## Purpose

Clear and well-defined



## Document Questions

Prepare a questionnaire in advance



## Define Population of Interest

Select population based on the purpose of analysis

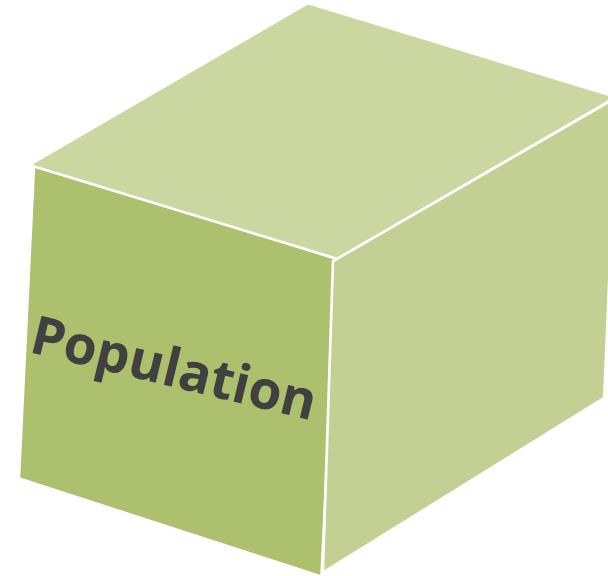


## Determine Sample

Based on the purpose of study

# **Population and Sample**

A population consists of various samples. The samples together represent the population.



A sample is:

- The part/piece drawn from the population
- The subset of the population
- A random selection to represent the characteristics of the population
- Representative analysis of the entire population

# Statistics and Parameters

“Statistics” are quantitative values calculated from the sample.

“Parameters” are the characteristics of the population.

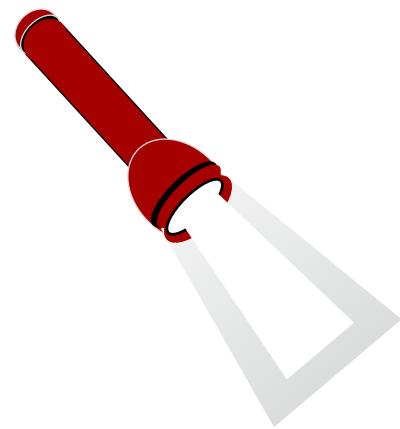
Sample →  $X_0, X_1, X_2, \dots, X_n$



	Population Parameters	Sample Statistics	Formula
Mean	$\mu$	$\bar{x}$	$\bar{x} = \frac{1}{n} \sum x_i$
Variance	$\sigma^2$	$S^2$	$S^2 = \frac{1}{n-1} \sum (x_i - \bar{x})^2$
Standard Deviation	$\sigma$	$S$	$S = \sqrt{\frac{1}{n-1} \sum (x_i - \bar{x})^2}$

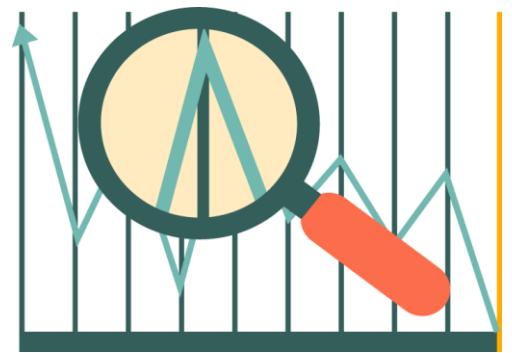
# Terms Used to Describe Data

Typical terms used in data analysis are:



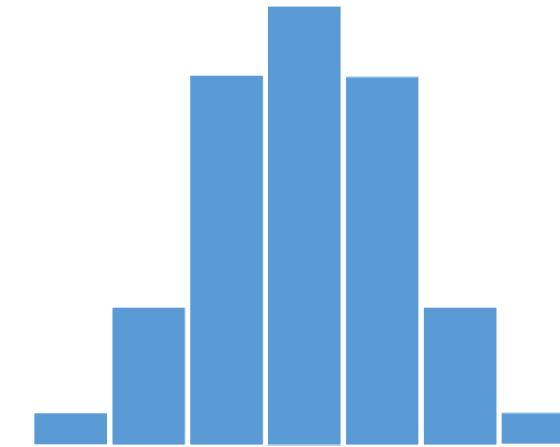
## SEARCH

“Search” is used to find unusual data. Data that does not match the parameters.



## INSPECT

“Inspect” refers to studying the shape and spread of data.



## CHARACTERIZE

“Characterize” refers to determining the central tendency of the data.



## CONCLUSION

“Conclusion” refers to preliminary or high-level conclusions about the data.

# Statistical Analysis Process

There are four steps in the statistical analysis process.

Step 1: Find the population of interest that suits the purpose of statistical analysis.

Step 2: Draw a random sample that represents the population.

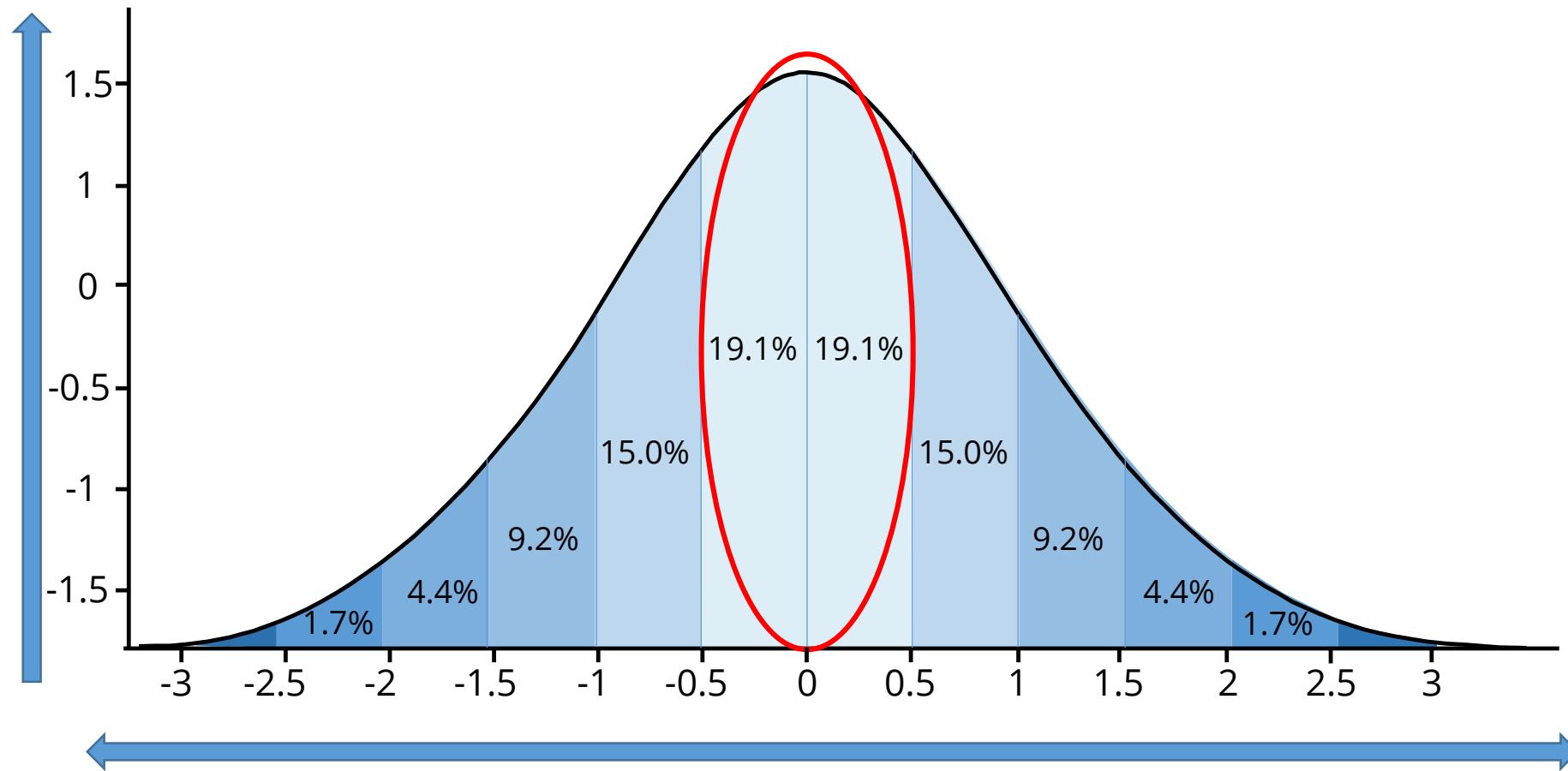
Step 3: Compute sample statistics to describe the spread and shape of the dataset.

Step 4: Make inferences using the sample and calculations. Apply it back to the population.



# Data Distribution

The collection of data values arranged in a sequence according to their relative frequency and occurrences.



**Range** of the data refers to minimum and maximum values.

**Frequency** indicates the number of occurrences of a data value.

**Central tendency** indicates data accumulation toward the middle of the distribution or toward the end.

# Measures of Central Tendency

The measures of central tendency are Mean, Median, and Mode.

**Mean** is the average.

Determine the mean score of these Math scores.

1. 80

2. 70

3. 75

4. 90

5. 80

6. 78

7. 55

8. 60

9. 80

$$\Sigma [80+70+75+90+80+78+55+60+80]/9$$

$$\text{Mean} = 74.22$$



**Median** is the 50<sup>th</sup> percentile.

55 60 70 75 78 80 80 80 90

$$\text{Median} = 78$$

**Mode** is the most frequent value.

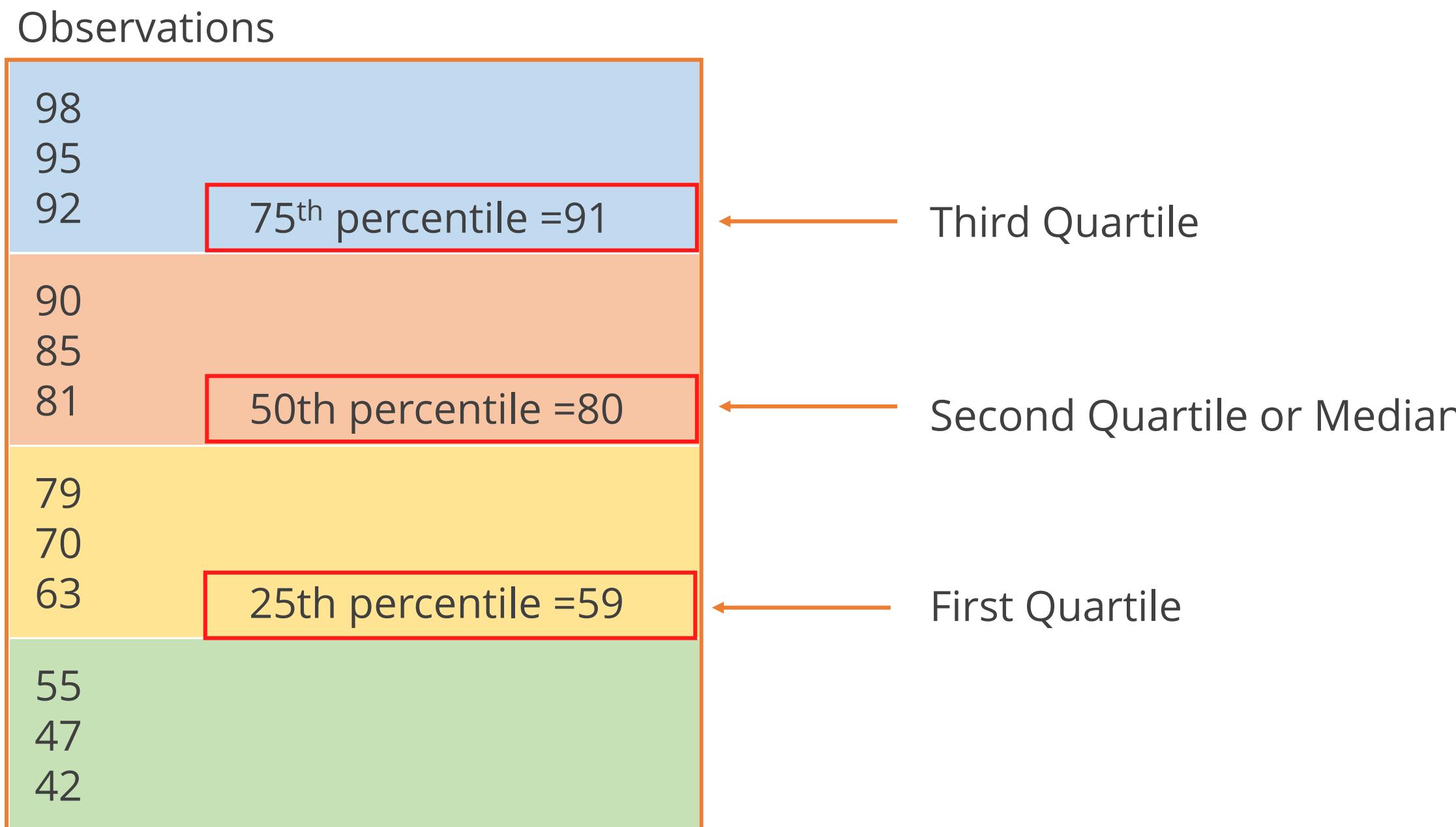
55 60 70 75 78 80 80 80 90

$$\text{Mode} = 80$$



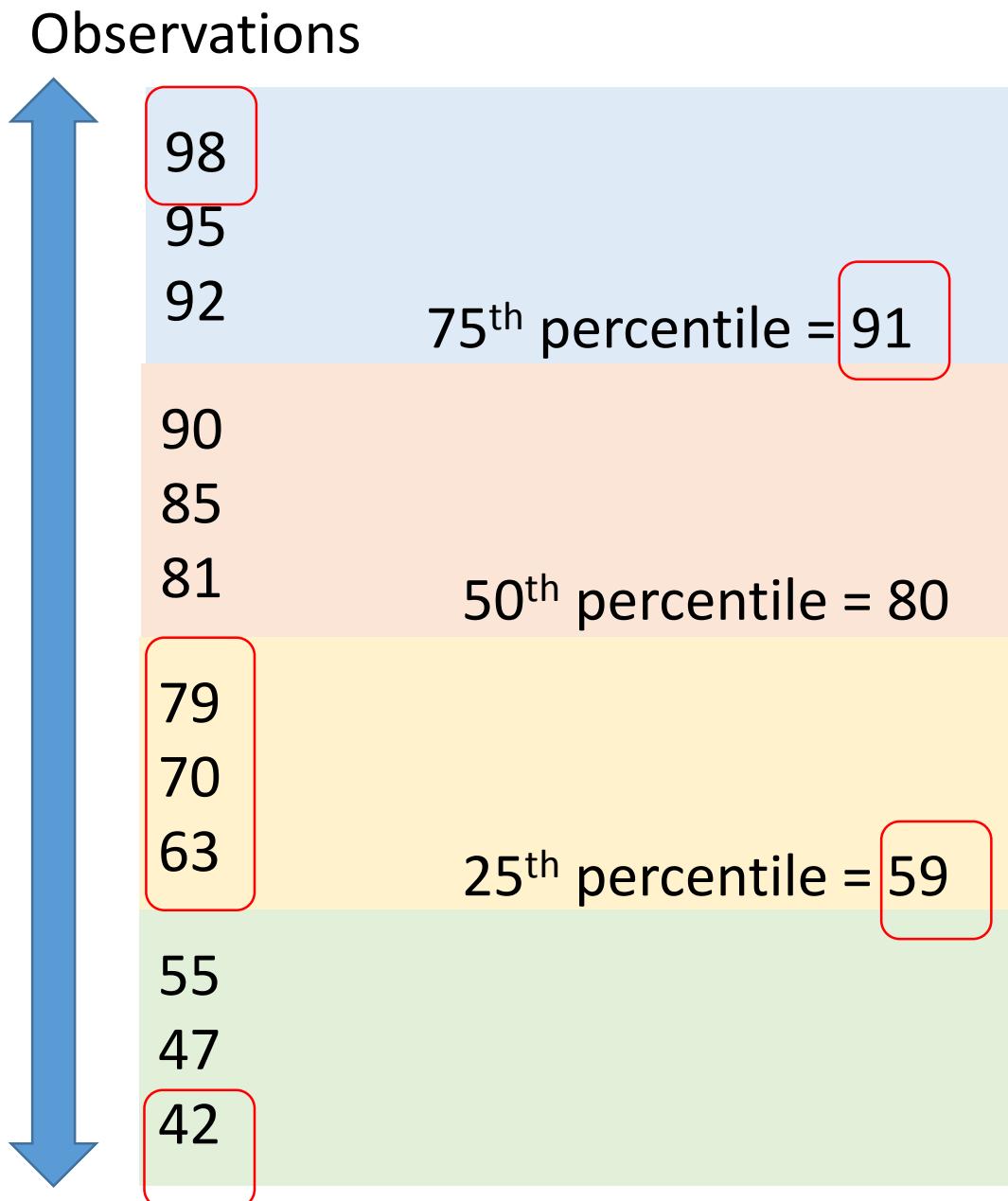
# Percentiles in Data Distribution

A percentile (or a centile) indicates the value below which a given percentage of observations fall.



# Dispersion

Dispersion denotes how stretched or squeezed a distribution is.



**Range:** The difference between the maximum and minimum values

**Inter-quartile Range:** Difference between the 25<sup>th</sup> and 75<sup>th</sup> percentiles

**Variance:** Data values around the Mean. (74.75)

**Standard Deviation:** Square root of the variance measured in small units



# Knowledge Check

KNOWLEDGE  
CHECK

What does frequency indicate?

- a. Range of the values present in the dataset
- b. Number of occurrences of a particular value in a dataset
- c. How spread out the data is
- d. Size of the sample drawn from a population



KNOWLEDGE  
CHECK

What does frequency indicate?

- a. Range of the values present in the dataset
- b. Number of occurrences of a particular value in a dataset
- c. How spread out the data is
- d. Size of the sample drawn from a population



The correct answer is **b**.

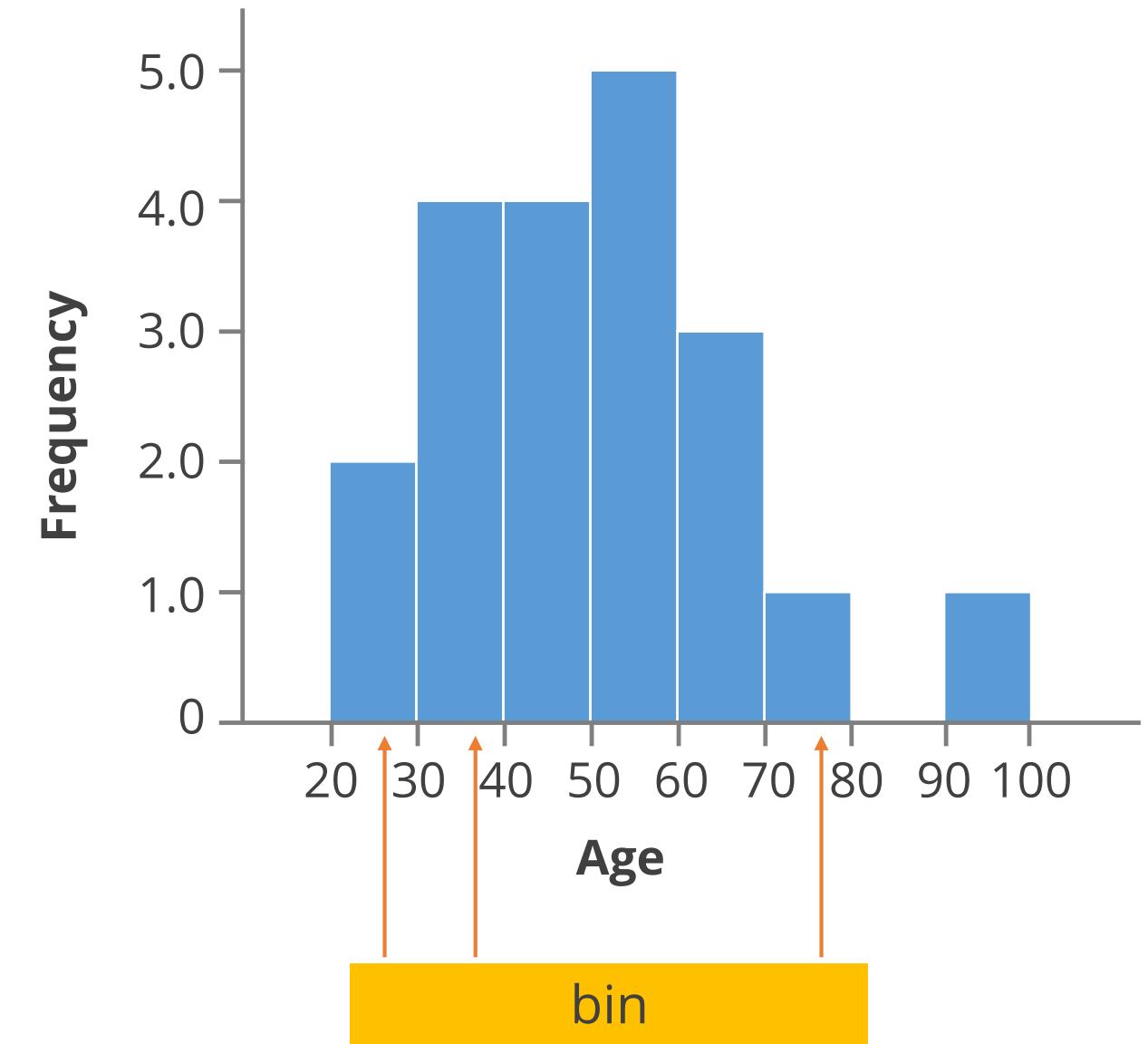
**Explanation:** Frequency indicates the number of occurrences of a particular value in a dataset.

# Histogram

Graphical representation of data distribution

## Features of a Histogram:

- It was first introduced by Karl Pearson.
- To construct a Histogram, “bin” the range of values.
- Bins are consecutive, non-overlapping intervals of a variable.
- Bins are of equal size.
- The bars represent the bins.
- The height of the bar represents the frequency of the values in the bin.
- It helps assess the probability distribution of a variable.

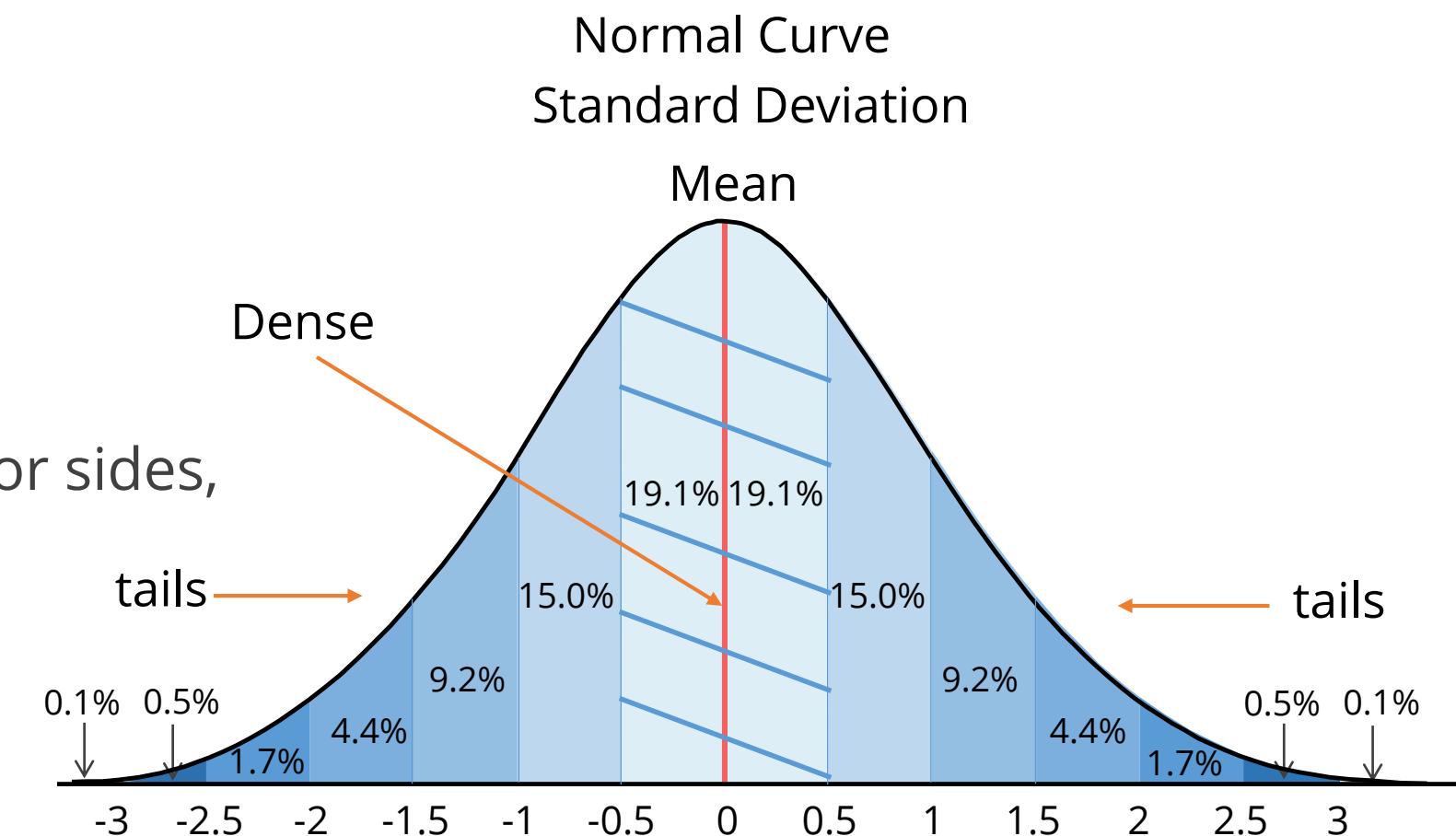


# Bell Curve – Normal Distribution

The bell curve is characterized by its bell shape and two parameters, mean and standard deviation.

## Bell curve is:

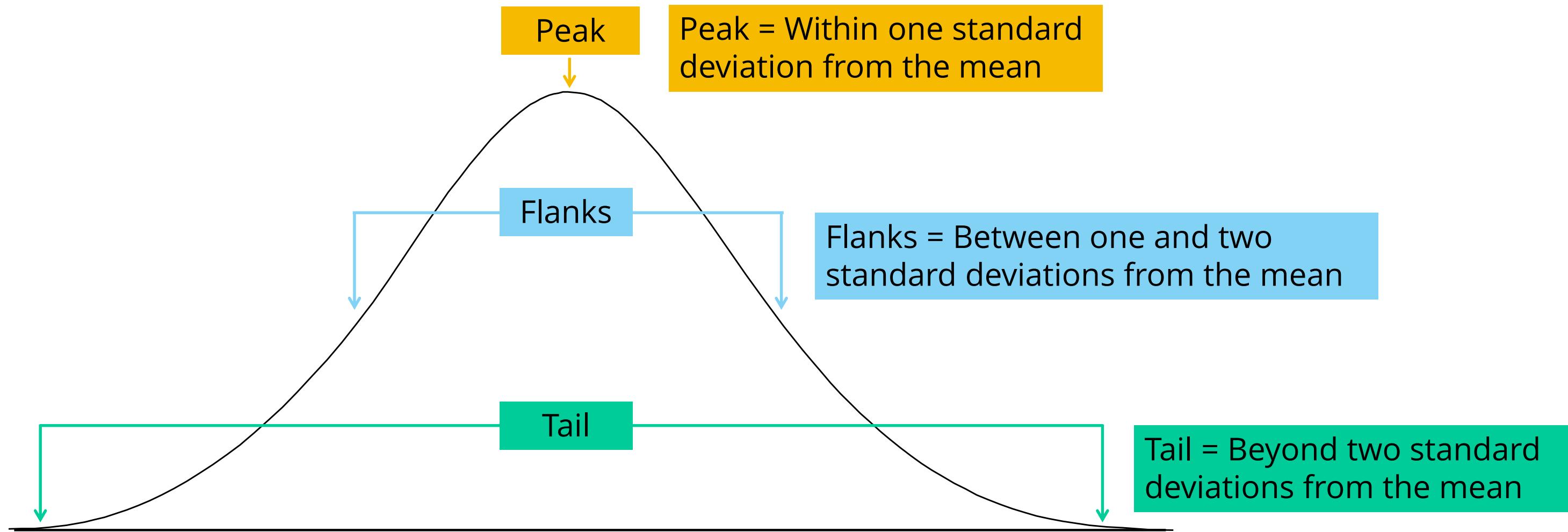
- Symmetric around the mean,
- Symmetric on both sides of the center,
- Having equal mean, median, and mode values,
- Denser in the center and less dense in the tails or sides,
- Defined by mean and standard deviation, and
- Known as the “Gaussian” curve.



The Bell curve is fully characterized by the mean ( $\mu$ ) and standard deviation ( $\sigma$ ).

# The Bell Curve

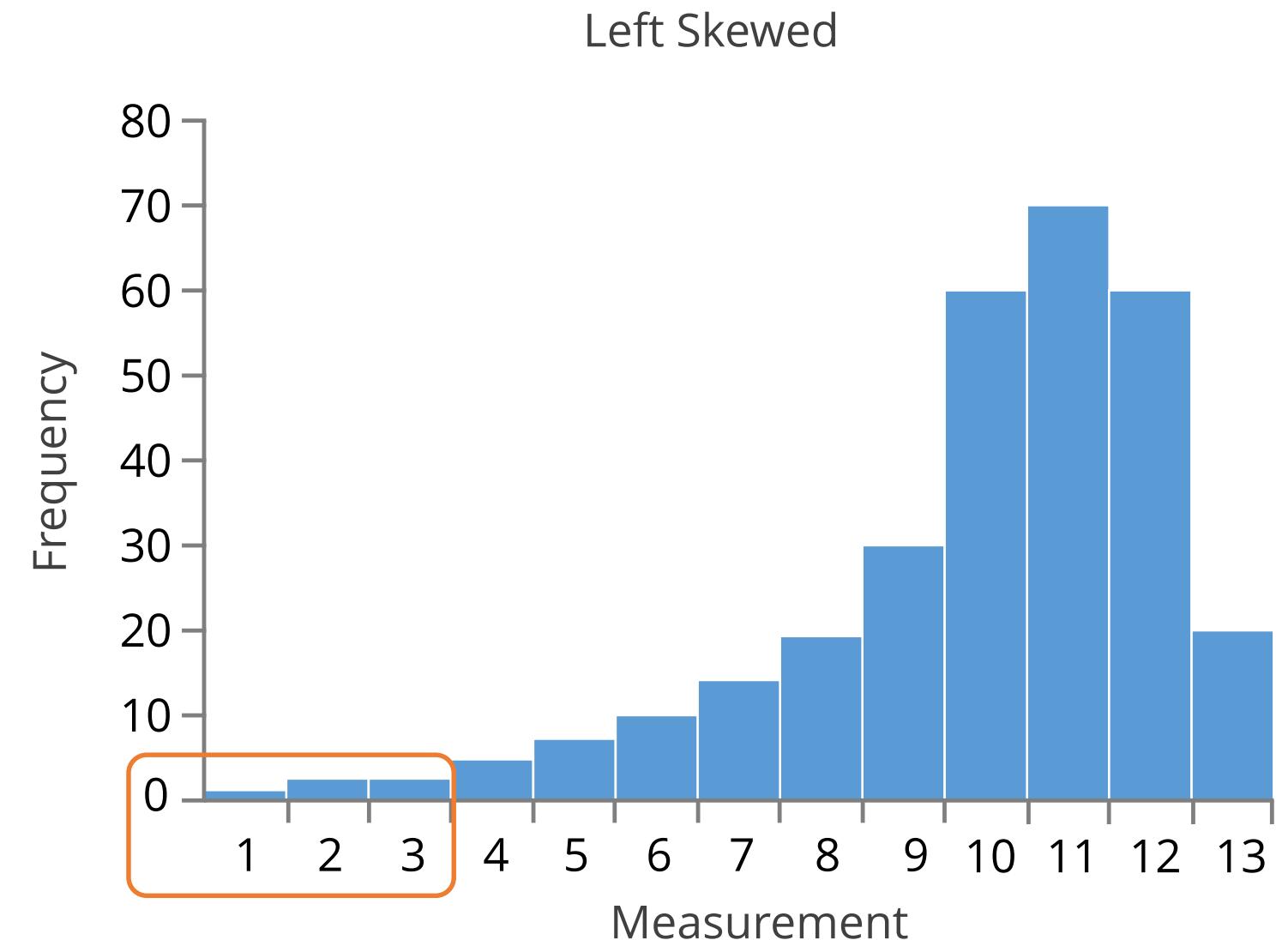
The Bell curve is divided into three parts to understand data distribution better.



# Bell Curve – Left Skewed

Skewed data distribution indicates the tendency of the data distribution to be more spread out on one side.

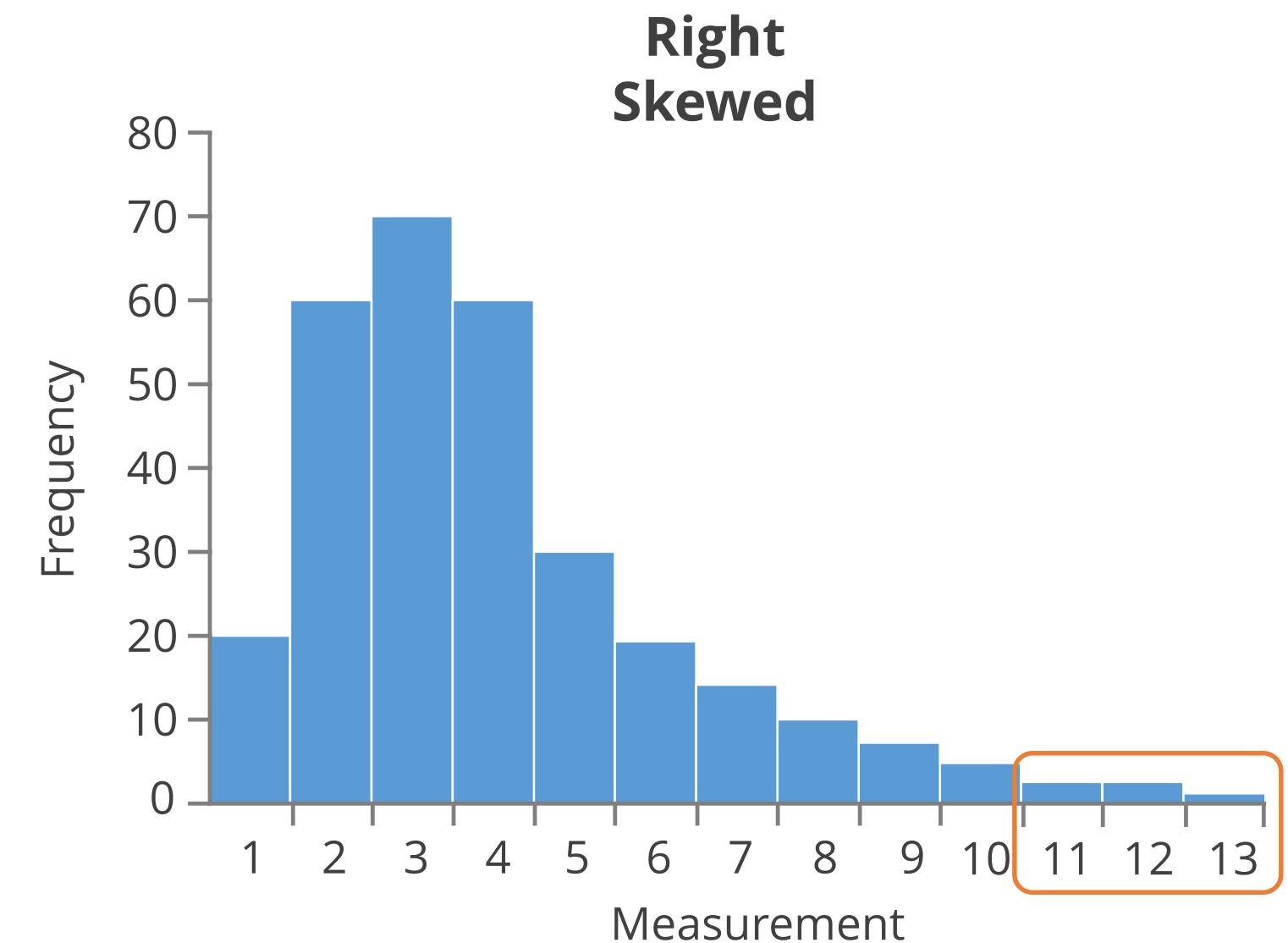
- The data is left skewed.
- Mean < Median
- The distribution is negatively skewed.
- Left tail contains large distributions.



## Bell Curve – Right Skewed

Skewed data distribution indicates the tendency of the data distribution to be more spread out on one side.

- The data is right skewed.
- The distribution is positively skewed.
- Mean > Median
- Right tail contains large distributions.



# Kurtosis

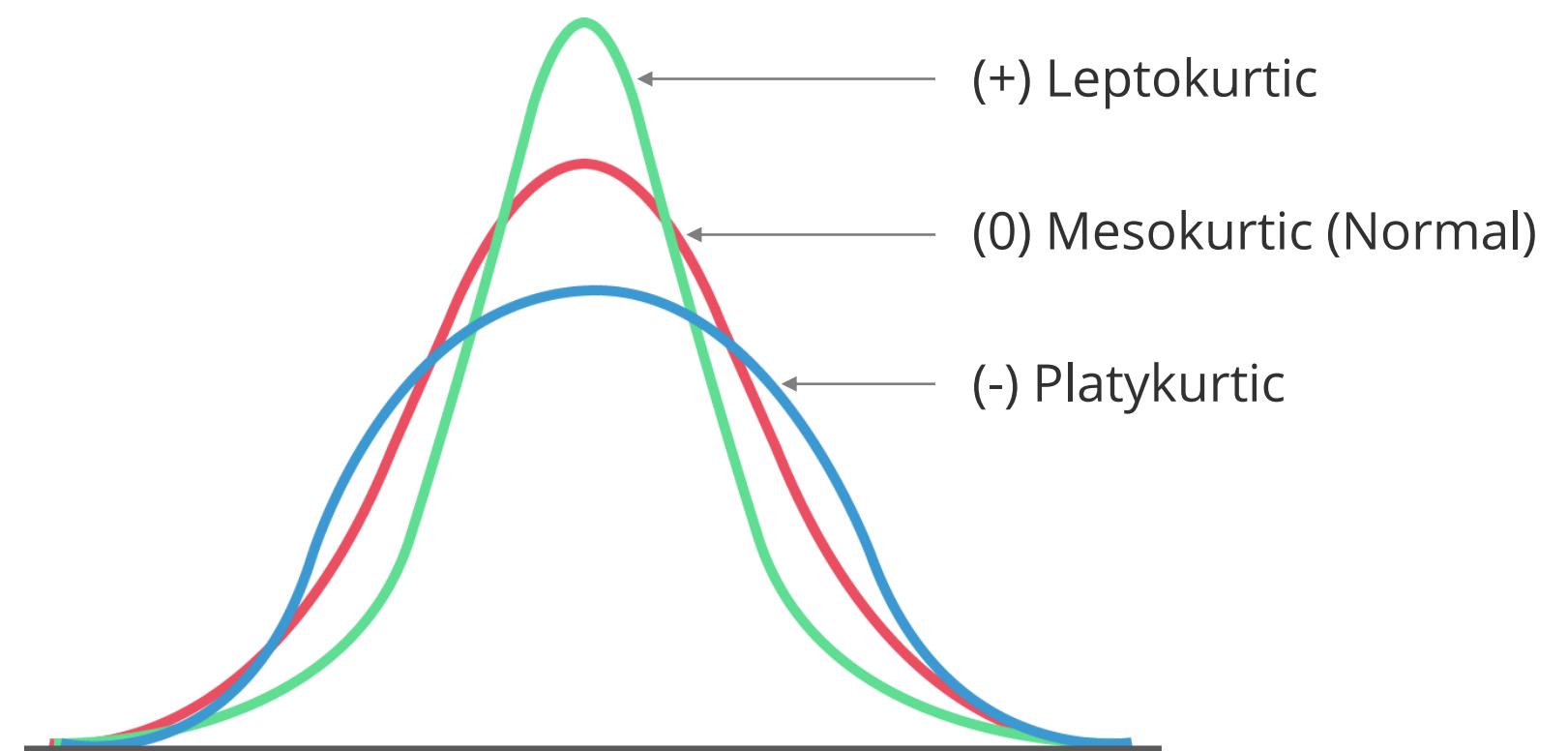
Kurtosis describes the shape of a probability distribution.

Kurtosis measures the tendency of the data toward the center or toward the tail.

**Platykurtic** is negative kurtosis.

**Mesokurtic** represents a normal distribution curve.

**Leptokurtic** is positive kurtosis.





# Knowledge Check

KNOWLEDGE  
CHECK

Which of the following is true for a normal distribution?

- a. Mean and median are equal
- b. Mean and mode are equal
- c. Mean, median, and mode are equal
- d. Mode and median are equal



KNOWLEDGE  
CHECK

Which of the following is true for a normal distribution?

- a. Mean and median are equal
- b. Mean and mode are equal
- c. Mean, median and mode are equal
- d. Mode and median are equal



The correct answer is

· c

**Explanation:** for Bell curve mean, median, and mode are equal.

# Hypothesis Testing

Hypothesis testing is an inferential statistical technique that determines if a certain condition is true for the population.

Alternative Hypothesis ( $H_1$ )	Null Hypothesis ( $H_0$ )
A statement that has to be concluded as true.	A statement of “no effect” or “no difference”.
It's a research hypothesis.	It's the logical opposite of the alternative hypothesis.
It needs significant evidence to support the initial hypothesis.	It indicates that the alternative hypothesis is incorrect.
If the alternative hypothesis garners strong evidence, reject the null hypothesis.	Weak evidence of alternative hypothesis indicates that the null hypothesis has to be accepted.

# Hypothesis Testing – Error Types

Representation of decision parameters using null hypothesis

## Type I Error ( $\alpha$ )

- Rejects the null hypothesis when it is true
- The probability of making Type I error is represented by  $\alpha$

## Type II Error ( $\beta$ )

- Fails to Reject the null hypothesis when it false
- The probability of making Type II error is represented by  $\beta$

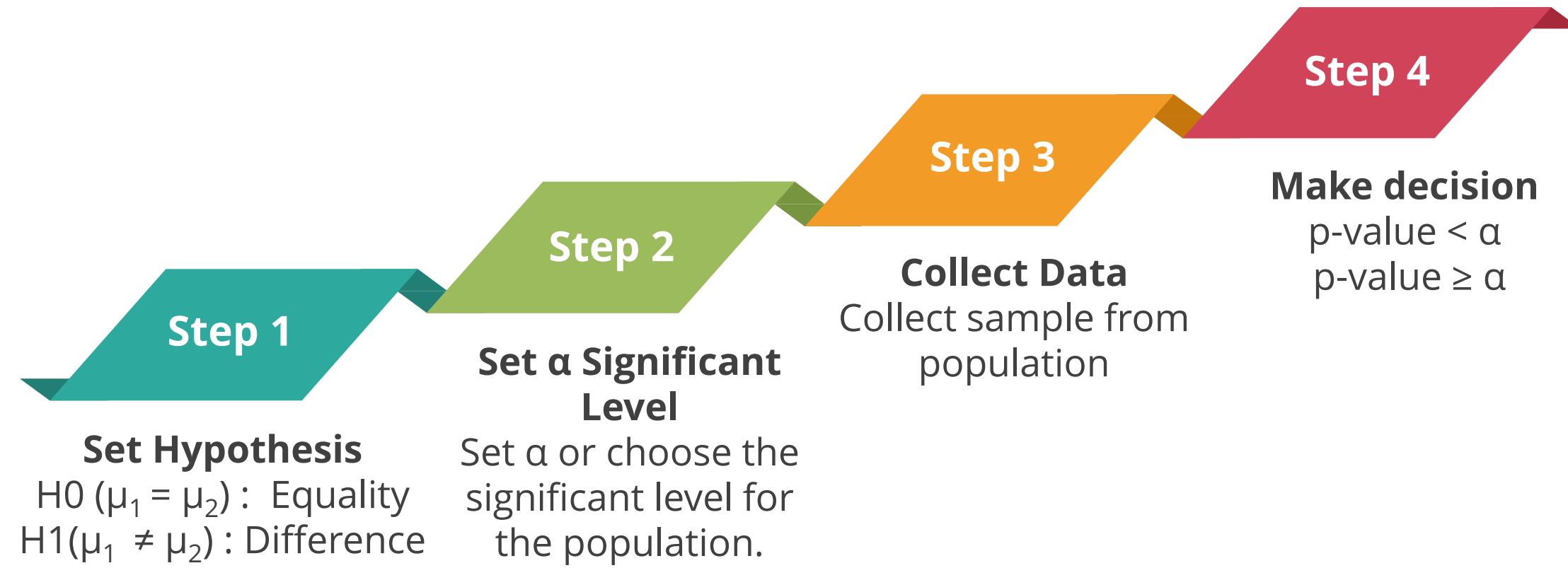
## *p-value*

- The probability of observing extreme values
- Calculated from collected data

Decision	Ho is True	Ho is False
Fail to Reject Null	Correct	Type II Error
Reject Null	Type I Error	Correct

# Hypothesis Testing - Process

There are four steps to the hypothesis testing process.



Reject the null hypothesis if  $p\text{-value} < \alpha$   
Fail to reject the null hypothesis if  $p\text{-value} \geq \alpha$

# Perform Hypothesis Testing

An example of clinical trials data analysis.



Company A



Company B

Null Hypothesis:  
Both medicines are  
equally effective.



Alternative Hypothesis:  
Both medicines are  
NOT equally effective.

# Data for Hypothesis Testing

There are three types of data on which you can perform hypothesis testing.



## Continuous Data

Evaluate the mean, median, standard deviation, or variance.



## Binomial Data

Evaluate the percentage, general classification of data.



## Poisson Data

Evaluate rate of occurrence or frequency.

# Types of Variables

There are three types of variables in categorical data.



## Nominal Variables

- Values with no logical ordering
- Variables are independent of each other
- Sequence does not matter



## Ordinal Variables

- Values are in logical order
- Relative distance between two data values is not clear

## Association

Two variables are associated or independent of each other:



Train A		Train B	
85%	15%	68%	32%
85%	15%	95%	55%

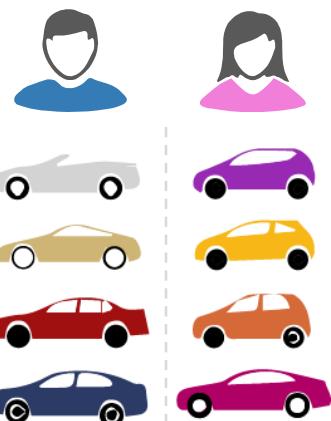
# Chi-Square Test

It is a hypothesis test that compares the observed distribution of your data to an expected distribution of data.



## Test of Association:

To determine whether one variable is associated with a different variable. For example, determine whether the sales for different cellphones depends on the city or country where they are sold.



## Test of Independence:

To determine whether the observed value of one variable depends on the observed value of a different variable. For example, determine whether the color of the car that a person chooses is independent of the person's gender.



Test is usually applied when there are two categorical variables from a single population.

# Chi Square Test - Example

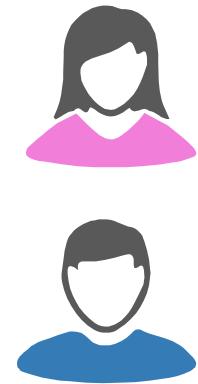
An example of Chi-Square test.

## Null Hypothesis

- There is no association between gender and purchase.
- The probability of purchase does not change for 500 dollars or more whether female or male.

## Alternative Hypothesis

- There is association between gender and purchase.
- The probability of purchase over 500 dollars is different for female and male.



	<\$500	>\$500
fo	.55	.45
fo	.75	.25

# Types of Frequencies

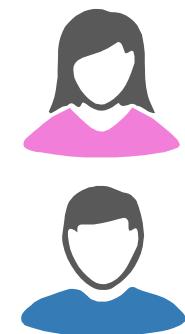
Expected and observed frequencies are the two types of frequencies.

## Expected Frequencies (fe)

The cell frequencies that are expected in a bivariate table if the two tables are statistically independent.

## Observed Frequencies (fo)

- There is association between gender and purchase.
- The probability of purchase over 500 dollars is different for female and male.



	Purchases	
	<\$500	>\$500
fo	.55	.45
fo	.75	.25

### No Association

Observed Frequency = Expected Frequency

### Association

Observed Frequency  $\neq$  Expected Frequency

# Features of Frequencies

---

The formula for calculating expected and observed frequencies using Chi Square:

$$\sum \frac{(f_e - f_o)^2}{f_e}$$

Features of Expected and Observed frequencies:

- Requires no assumption of the underlying population
- Requires random sampling



# Knowledge Check

KNOWLEDGE  
CHECK

In Chi-Square test, there is no association of variables if \_\_\_\_.

- a. Observed Frequency  $\neq$  Expected Frequency
- b. Observed Frequency = Expected Frequency
- c. Independent of observed frequencies
- d. Independent of expected frequencies



KNOWLEDGE  
CHECK

In Chi-Square test, there is no association of variables if:

- a. Observed Frequency  $\neq$  Expected Frequency
- b. Observed Frequency = Expected Frequency
- c. Independent of observed frequencies
- d. Independent of expected frequencies

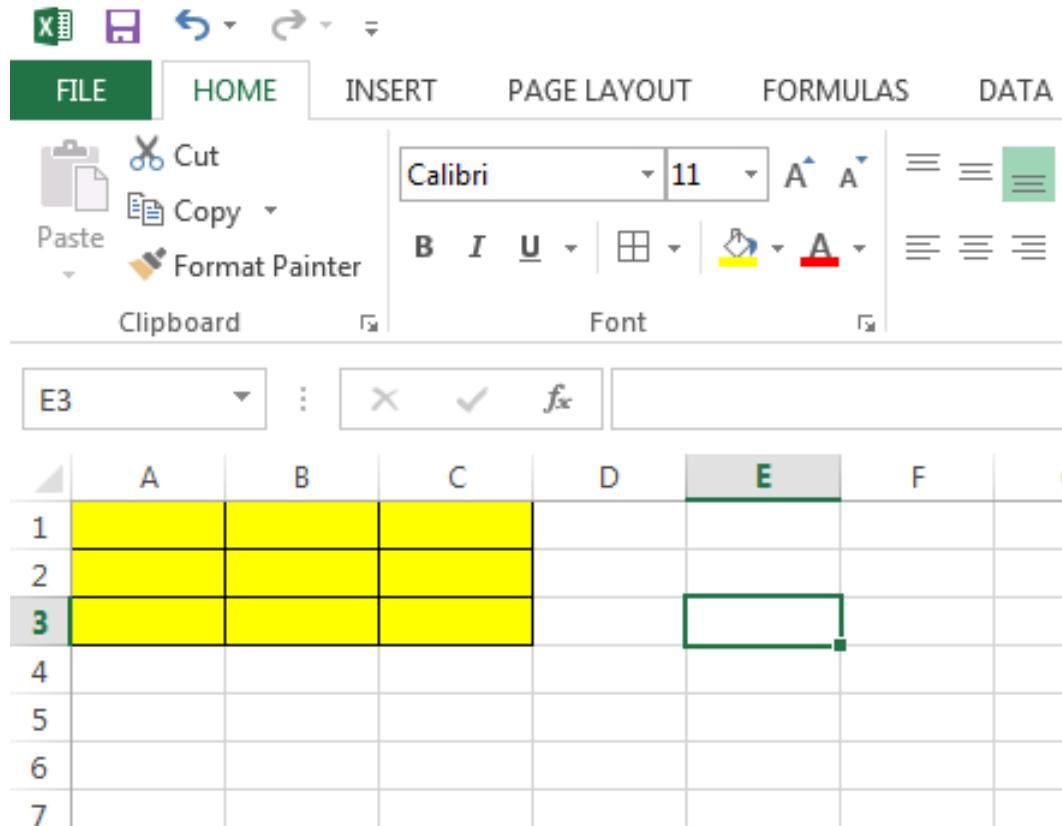


The correct answer is **b**.

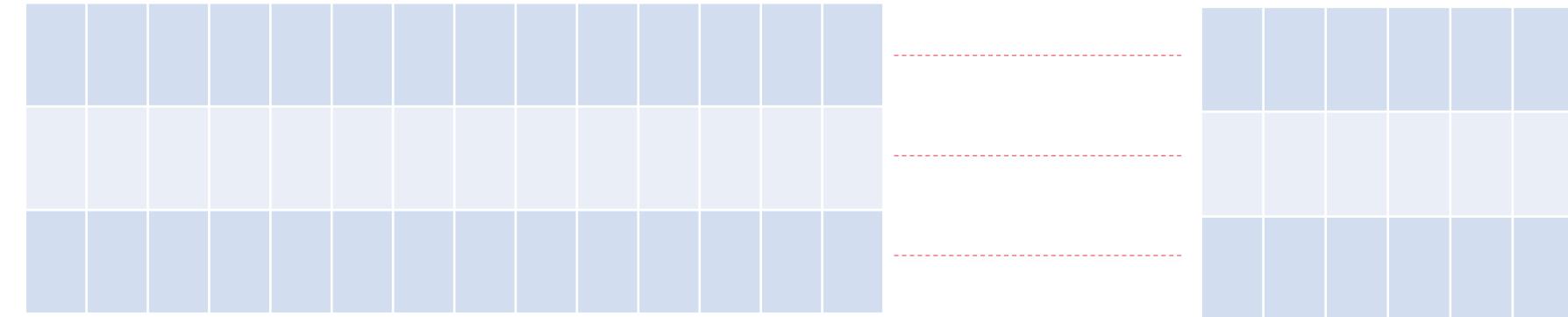
**Explanation:** Observed Frequency = Expected Frequency indicates no association.

# Correlation Matrix

A Correlation matrix is a square matrix that compares a large number of variables.



The screenshot shows a Microsoft Excel spreadsheet with a 3x3 matrix highlighted in yellow. The matrix consists of three rows and three columns labeled A, B, and C. The first row contains values 1, 2, and 3. The second row contains values 4, 5, and 6. The third row contains values 7, 8, and 9. The matrix is located in cells A1 to C3. The Excel ribbon is visible at the top, showing tabs like FILE, HOME, INSERT, PAGE LAYOUT, FORMULAS, and DATA. The font and style tools are also visible in the ribbon.



Correlation matrix – a square matrix

$n \times n$  Matrix

(very large number of rows and columns)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

$3 \times 3$  matrix (simple square matrix)

**Correlation coefficient** measures the extent to which two variables tend to change together. The coefficient describes both the strength and direction of the relationship.

# Correlation Matrix

A Correlation matrix is a square matrix that compares a large number of variables.

## Pearson product moment correlation

It evaluates the linear relationship between two continuous variables.

Linear relationship means that a change in one variable results in a proportional change in the other.

## Spearman rank order correlation

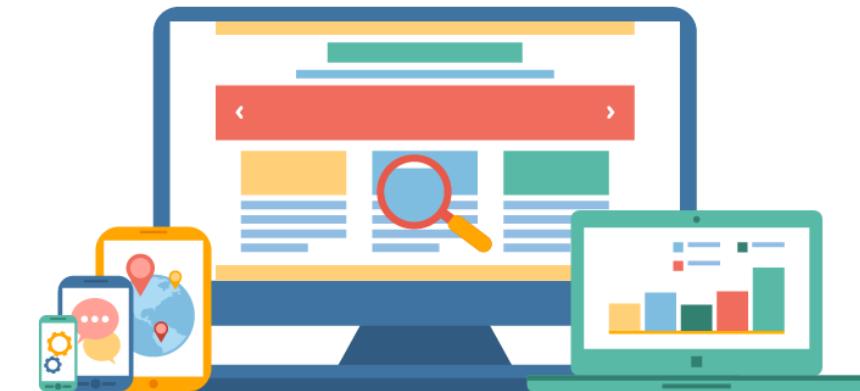
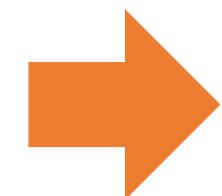
It evaluates the monotonic relationship between two continuous or ordinal variables.

- Monotonic relationship means that the variables tend to change together though not necessarily at a constant rate.
- The correlation coefficient is based on the ranked values for each variable rather than the raw data.

# Correlation Matrix - Example

An example of a correlation matrix calculated for a stock market.

	T	U	V	W	X	Y	Z
8	Correlation	EQUITY 1	EQUITY 2	FX FORWARD 1	FX FORWARD 2	BOND 1	BOND 2
9	EQUITY 1	1.00	0.38	0.20	0.45	- 0.17	- 0.12
10	EQUITY 2	0.38	1.00	0.54	0.51	- 0.20	0.12
11	FX FORWARD 1	0.20	0.54	1.00	0.35	- 0.14	0.16
12	FX FORWARD 2	0.45	0.51	0.35	1.00	- 0.11	- 0.09
13	BOND 1	- 0.17	- 0.20	- 0.14	- 0.11	1.00	0.03
14	BOND 2	- 0.12	0.12	0.16	- 0.09	0.03	1.00



A correlation matrix that is calculated for the stock market will probably show the short-term, medium-term, and long-term relationship between data variables.

# Inferential Statistics

Inferential statistics uses a random sample from the data to make inferences about the population.



Inferential statistics can be used only under the following conditions:

- A complete list of the members of the population is available.
- A random sample has been drawn from the population.
- Using a pre-established formula, you determine that the sample size is large enough.

Inferential statistics can be used even if the data does not meet the criteria.

- It can help determine the strength of the relationships within the sample.
- If it is very difficult to obtain a population list and draw a random sample, do the best you can with what you have.

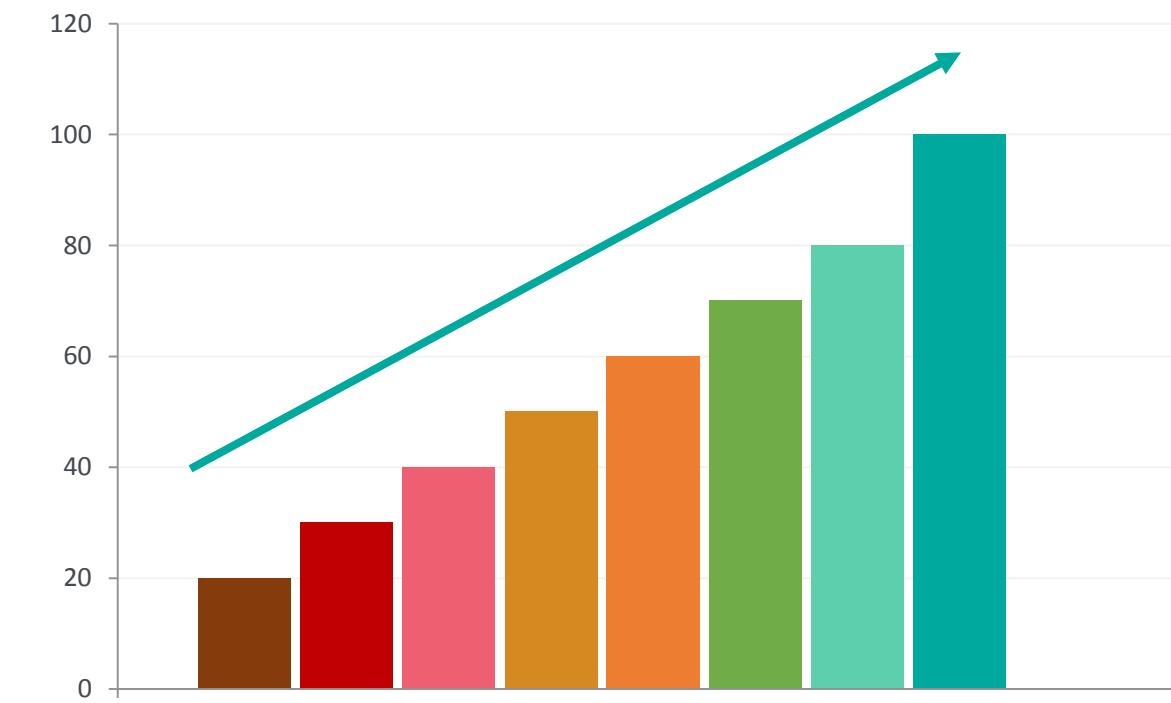
# Applications of Inferential Statistics

---

Inferential Statistics has its uses in almost every field such as business, medicine, data science, and so on.

## Inferential Statistics

- Is an effective tool for forecasting.
- Is used to predict future patterns.





**QUIZ**  
**1**

**If a sample of five boxes weigh 90, 135, 160, 115, and 110 pounds, what will be the median weight of this sample?**

- a. 160
- b. 115
- c. 90
- d. 135



**QUIZ**  
**1**

**If a sample of five boxes weigh 90, 135, 160, 115, and 110 pounds, what will be the median weight of this sample?**

- a. 160
- b. 115
- c. 90
- d. 135



The correct answer is **b**.

**Explanation:** Arrange in a sequential order and the middle number will be the median. If the set of numbers is even then take the average or mean of the two numbers in the middle.

**QUIZ**  
**2**

**Identify the parameters that characterize a bell curve. *Select all that apply.***

- a. Variance
- b. Mean
- c. Standard deviation
- d. Range



**QUIZ**  
**2**

**Identify the parameters that characterize a bell curve. *Select all that apply.***

- a. Variance
- b. Mean
- c. Standard deviation
- d. Range



The correct answer is **b,c**.

**Explanation:** Bell Curve is completely characterized by mean and standard deviation.

**QUIZ**  
**3**

**Identify the accurate statement about the relationship between standard deviation and variance.**

- a. Standard deviation is the square root of variance.
- b. Variance is the square root of standard deviation.
- c. Both are inversely proportional.
- d. Both are directly proportional.



**QUIZ**  
**3**

## Identify the accurate statement about the relationship between standard deviation and variance

- a. Standard deviation is the square root of variance.
- b. Variance is the square root of standard deviation.
- c. Both are inversely proportional.
- d. Both are directly proportional.



The correct answer is **a.**

**Explanation:** Standard deviation is the square root of variance.

**QUIZ****4**

**Identify the hypothesis decision rules. *Select all that apply.***

- a. Reject the null hypothesis if  $p\text{-value} < \alpha$
- b. Is independent of  $p\text{-value}$
- c. Fail to reject the null hypothesis if  $p\text{-value} \geq \alpha$
- d. Is independent of  $\alpha$



**QUIZ****4**

**Identify the hypothesis decision rules. *Select all that apply.***

- a. Reject the null hypothesis if  $p\text{-value} < \alpha$
- b. Is independent of  $p\text{-value}$
- c. Fail to reject the null hypothesis if  $p\text{-value} \geq \alpha$
- d. Is independent of  $\alpha$



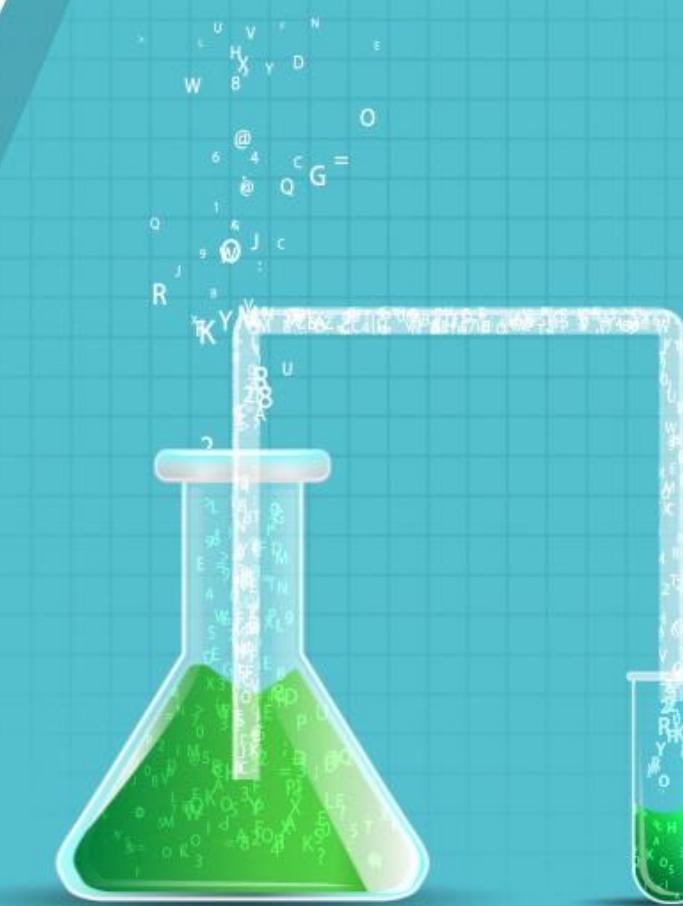
The correct answer is **a, c.**

**Explanation:** A hypothesis decision rule :

- Reject the null hypothesis if  $p\text{-value} < \alpha$
- Fail to reject the null hypothesis if  $p\text{-value} \geq \alpha$

# Key Takeaways

- Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data.
  - Statistical analysis is more reliable when compared to non-statistical analysis.
  - Descriptive and inferential are the two major categories of statistics.
  - Mean, median, and mode are measures of central tendency, while variance and standard deviation measure the spread of data.
  - The spread of distribution is called dispersion and is graphically represented by a histogram and a bell curve.
  - Hypothesis testing is an inferential statistical technique that is useful for forecasting future patterns.
  - Chi-Square test is a hypothesis test that compares observed distribution to an expected distribution.
  - The correlation coefficient or covariance is measured with the help of correlation matrix.



**This concludes “Statistical Analysis and Business  
Applications”**  
The next lesson is Data Analytics Overview

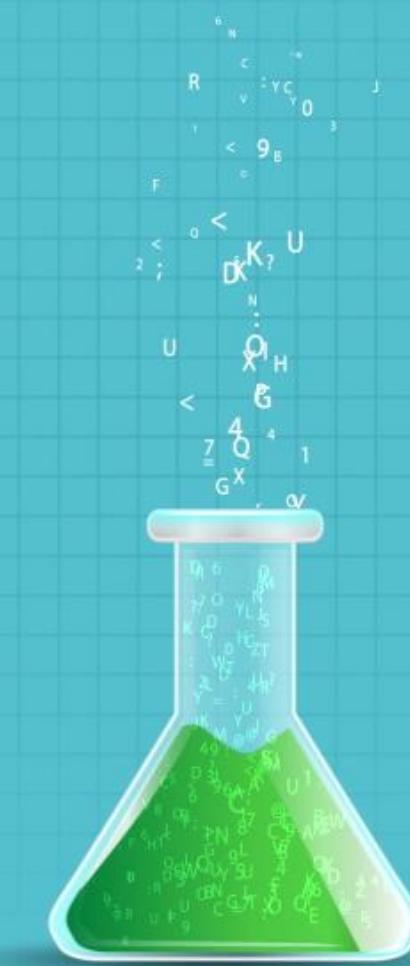
# Data Science with Python

## Lesson 04—Python: Environment Setup and Essentials



# What You'll Learn

- How to install Anaconda and Jupyter notebook
  - Some of the important data types supported by Python
  - Data structures such as lists, tuples, sets, and dicts
  - Slicing and accessing the four data structures
  - Few basic operators and functions
  - Some important control flow statements



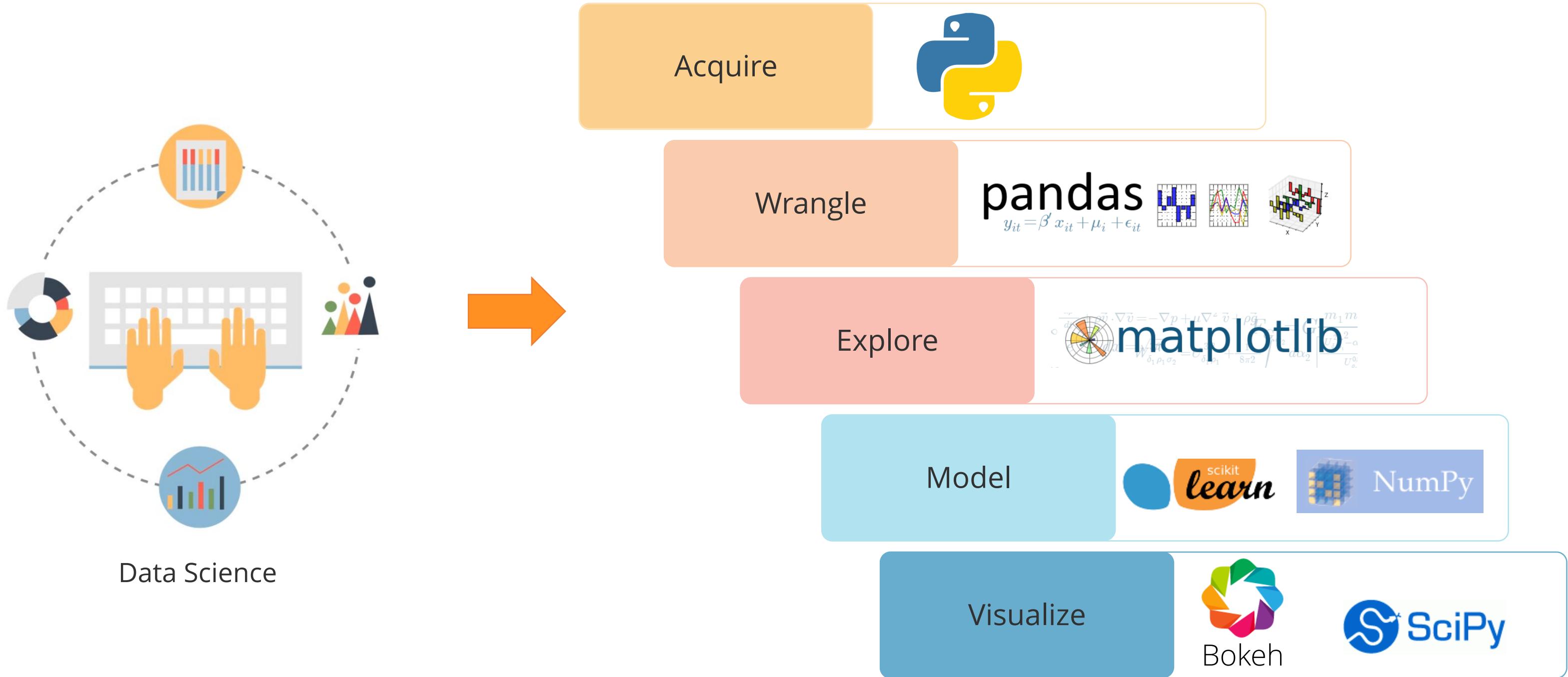
# Python: Environment Setup and Essentials

## Python Environment Setup

DATA  
SCIENCE

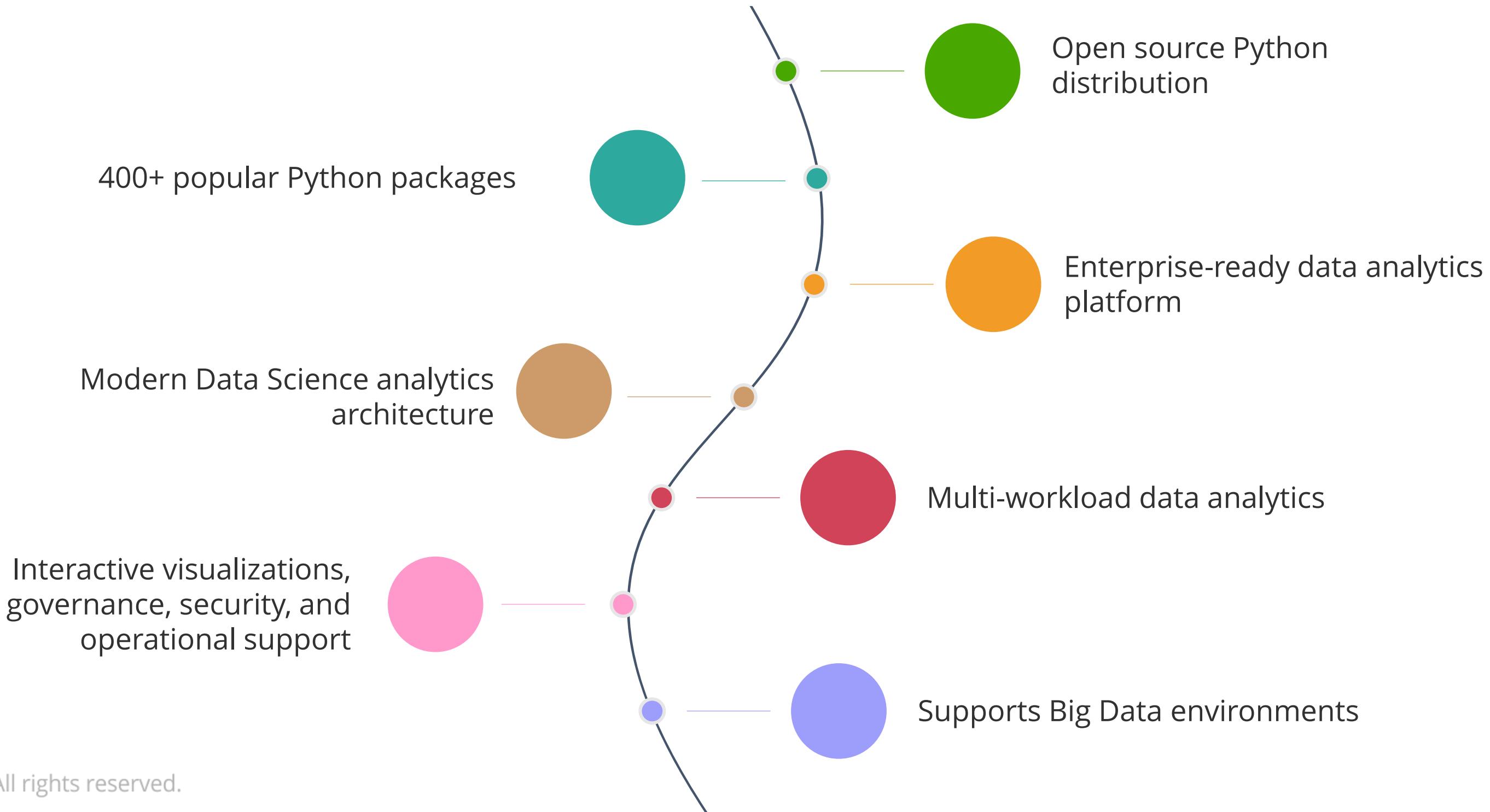
# Quick Recap: Python for Data Science

You have seen how Python and its different libraries are useful in various aspects of Data Science.



# Why Anaconda

To use Python, we recommend that you download Anaconda. Following are some of the reasons why Anaconda is one of the best Data Science platforms:



# Installation of Anaconda Python Distribution

Currently, there are two versions of Python. You can download and use either of them although the 2.7 version is preferable.

PYTHON 2.7	PYTHON 3.5
WINDOWS 64-BIT GRAPHICAL INSTALLER	WINDOWS 64-BIT GRAPHICAL INSTALLER
335M	345M

<a href="#">Windows 32-bit Graphical Installer</a>	<a href="#">Windows 32-bit Graphical Installer</a>
281M	283M

# Installation of Anaconda Python Distribution (contd.)

You can install and run the Anaconda Python distribution on different platforms.

Windows

Mac OS

Linux

PYTHON 2.7

WINDOWS 64-BIT  
GRAPHICAL INSTALLER

335M

Windows 32-bit  
Graphical Installer

281M



**Website URL:**

<https://www.continuum.io/downloads>

**Graphical Installer**

- Download the graphical installer.
- Double-click the .exe file to install Anaconda and follow the instructions on the screen.

*Click each tab to know how to install Python on those operating systems.*

# Installation of Anaconda Python Distribution (contd.)

You can install and run the Anaconda Python distribution on different platforms.

Windows

Mac OS

Linux

PYTHON 2.7

MAC OS X 64-BIT  
GRAPHICAL INSTALLER

339M (OS X 10.7 or higher)

Mac OS X 64-bit  
Command-Line installer

290M (OS X 10.7 or higher)



**Website URL:**

<https://www.continuum.io/downloads>

## Graphical Installer

- Download the graphical installer.
- Double-click the downloaded .pkg file and follow the instructions.

## Command Line Installer

- Download the command line installer.
- In your terminal window, type the command listed below and follow the given instructions:

### Python 2.7:

`bash Anaconda2-4.0.0-MacOSX-x86_64.sh`

*Click each tab to know how to install Python on those operating systems.*

# Installation of Anaconda Python Distribution (contd.)

You can install and run the Anaconda Python distribution on different platforms.

Windows

Mac OS

Linux

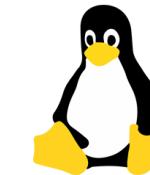
PYTHON 2.7

LINUX 64-BIT

392M

Linux 32-bit

332M



**Website URL:**

<https://www.continuum.io/downloads>

**Command Line Installer**

- Download the installer.
- In your terminal window, type the command line shown below and follow the instructions:

**Python 2.7:**

`bash Anaconda2-4.0.0-Linux-x86_64.sh`

*Click each tab to know how to install Python on those operating systems.*

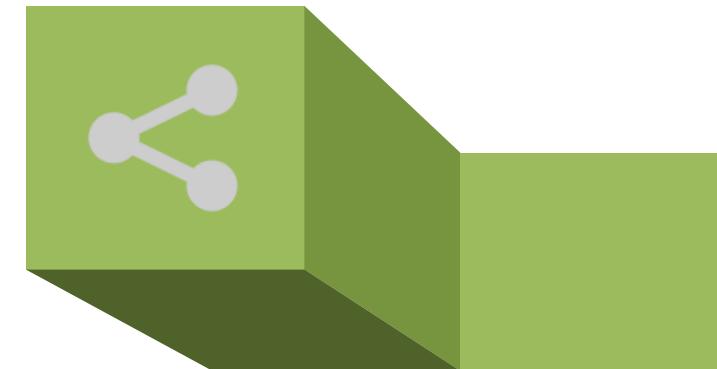
# Jupyter Notebook

Jupyter is an open source and interactive web-based Python interface for Data Science and scientific computing. Some of its advantages are:

Python language support



Content sharing and contribution



Big Data platform integration

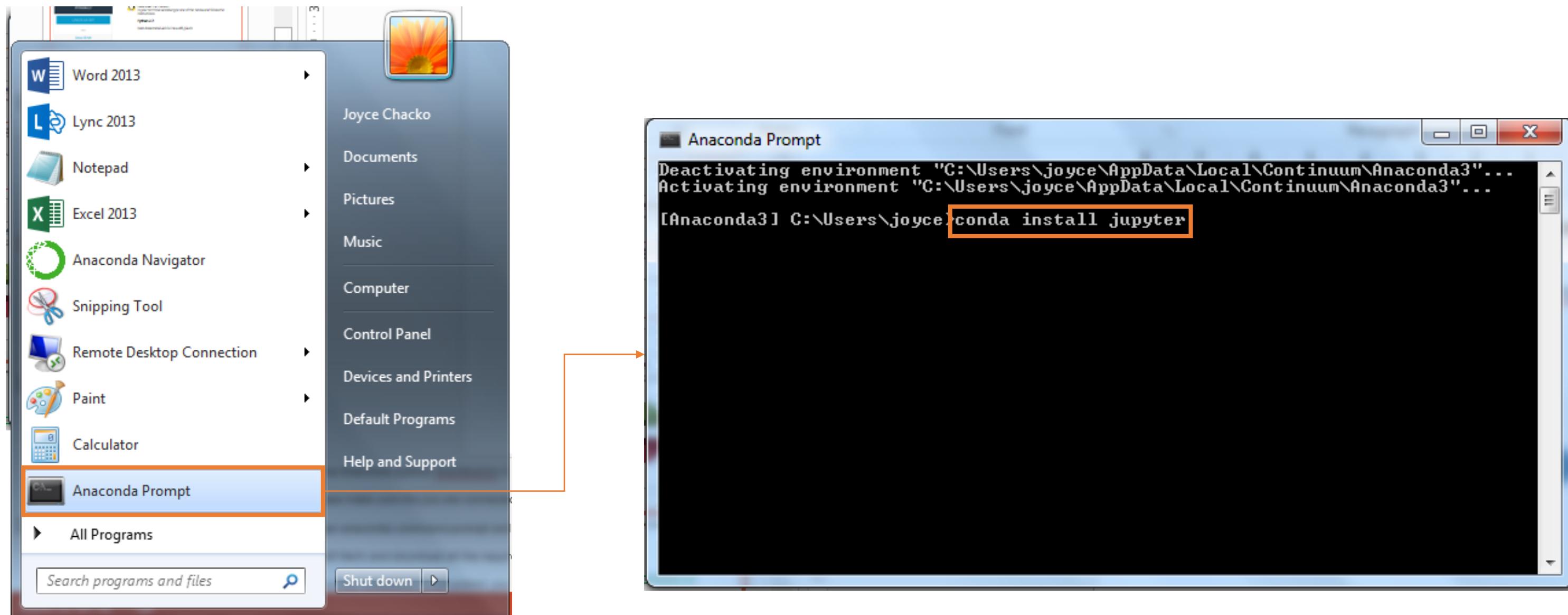


Built-in interactive widgets



# Jupyter Notebook: Installation

To install Jupyter notebook on your system, type the command shown here on Anaconda prompt and press Enter to execute it.



# Python: Environment Setup and Essentials

Python Primer

DATA  
SCIENCE

# Getting Started

The screenshot shows a Jupyter Notebook interface with the title "Basic Python". The URL in the browser bar is "localhost:8888/notebooks/Basic%20Python/Basic%20Python.ipynb". The notebook has a Python 2 kernel selected. The code cells and their outputs are as follows:

- In [1]: `import sys` Import sys module
- In [2]: `print sys.version` Print sys version  
2.7.11 |Anaconda 2.5.0 (64-bit)| (default, Feb 16 2016, 09:58:36) [MSC v.1500 64 bit (AMD64)]
- In [3]: `import platform` Import platform library
- In [4]: `platform.python_version()` View python version  
Out[4]: '2.7.11'
- In [5]: `# A Hello world example` Comment line  
`print 'hello world'` Test string  
Out[5]: hello world
- In [6]: `3+5` Test number operation  
Out[6]: 8
- In [7]: `8*4`  
Out[7]: 32

# Variables and Assignment

A variable can be assigned or bound to any value. Some of the characteristics of binding a variable in Python are listed here:

```
In [1]: x = 3  
        type(x)
```

The variable refers to the memory location of the assigned value.

Out[1]: int

```
In [2]: y = 2.1  
        type(y)
```

The variable appears on the left, while the value appears on the right.

Out[2]: float

```
In [3]: z = 'test'  
        type(z)
```

The data type of the assigned value and the variable is the same.

Out[3]: str

# Example—Variables and Assignment

Let us look at an example of how you can assign a value to a variable, and print it and its data type.

```
In [44]: first_string_variable = 'test'  
        first_integer_variable = 123
```

Assignment

```
In [45]: print first_string_variable  
        print first_integer_variable
```

```
test  
123
```

Variable data value

```
In [47]: print type(first_string_variable)  
        print type(first_integer_variable)
```

```
<type 'str'>  
<type 'int'>
```

Data type of the object

# Multiple Assignments

You can access a variable only if it is defined. You can define multiple variables simultaneously.

In [48]: `number_example`

```
NameError Traceback (most recent call last)
<ipython-input-48-a856f233ae98> in <module>()
      1 number_example
NameError: name 'number_example' is not defined
```

Access variable without assignment

In [49]: `number_example = 2`

Out[49]: 2

In [54]: `integer_x, integer_y = 5, 22`

In [55]: `integer_x`

Out[55]: 5

In [56]: `integer_y`

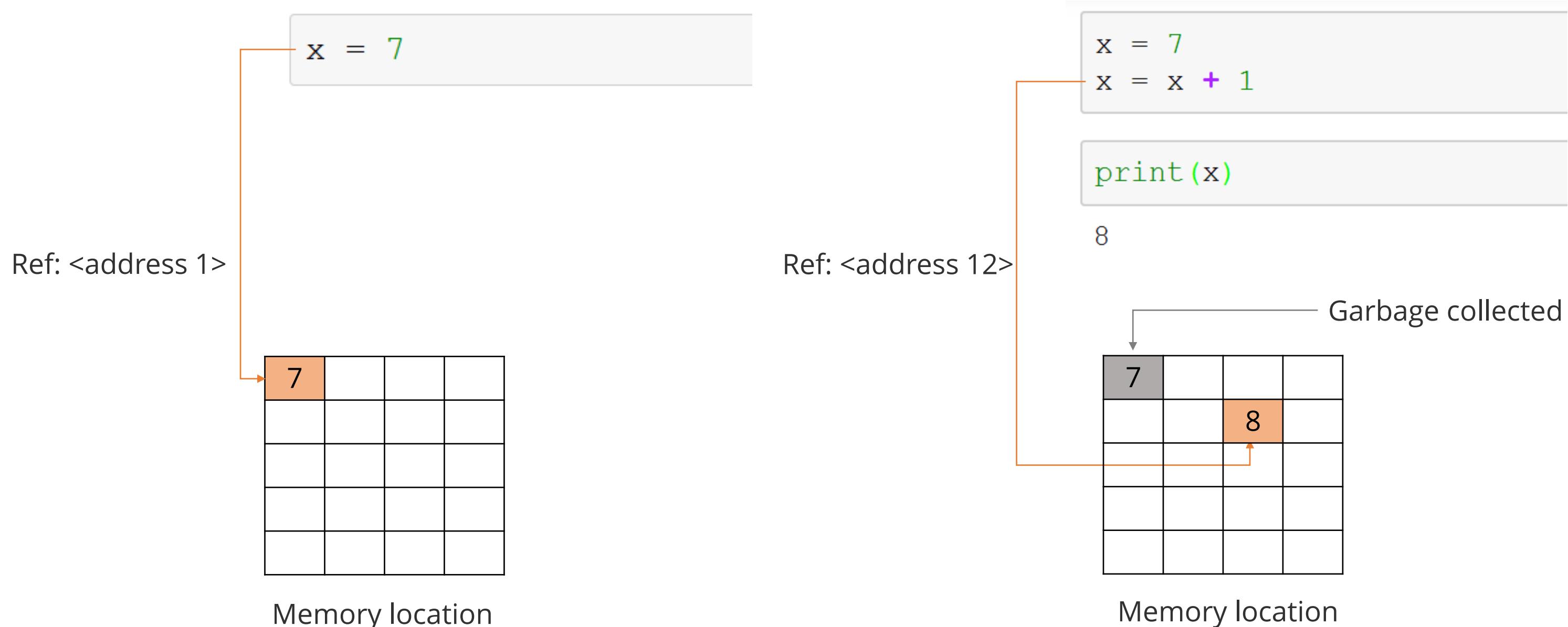
Out[56]: 22

Access variable after assignment

Multiple assignments

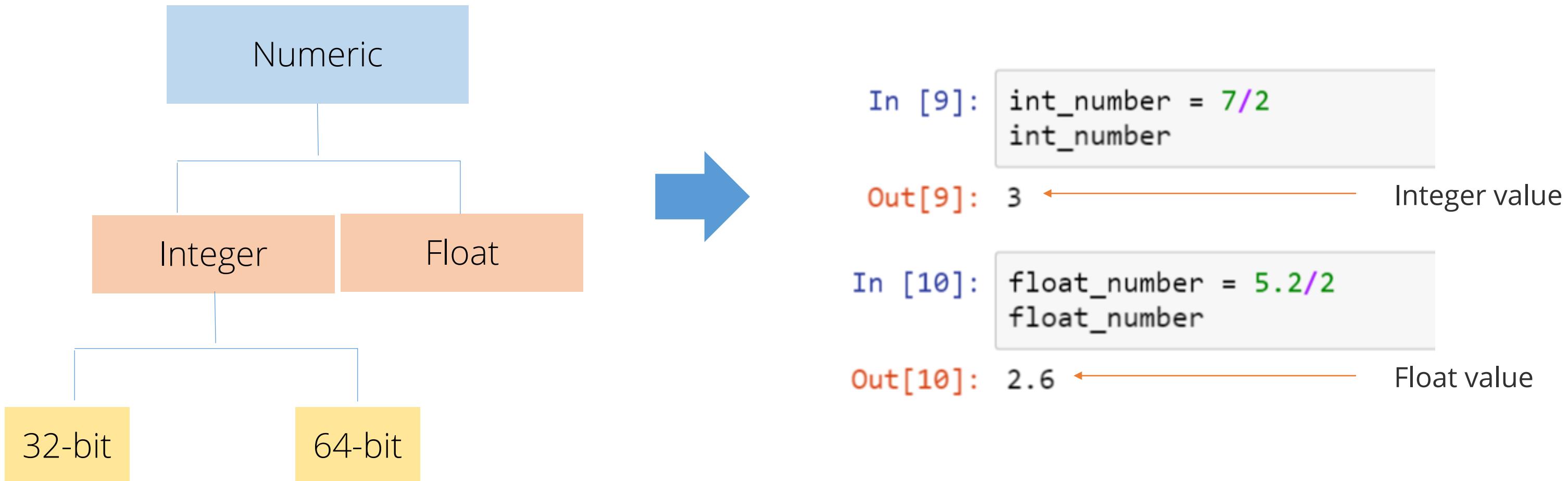
# Assignment and Reference

When a variable is assigned a value, it refers to the value's memory location or address. It does not equal the value itself.



# Basic Data Types: Integer and Float

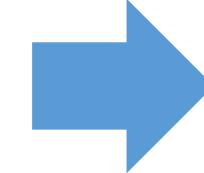
Python supports various data types. There are two main numeric data types:



# Basic Data Types: String

Python has extremely powerful and flexible built-in string processing capabilities.

String



In [14]:

```
string_one = 'first string'  
string_two = "second string"  
string_three = """third string"""
```

With single quote

With double quote

Three double quotes

In [15]:

```
print string_one  
print string_two  
print string_three
```

Print string values

```
first string  
second string  
third string
```

# Basic Data Types: None and Boolean

Python also supports the Null and Boolean data types.

```
In [102]: num_x = None           Null value type  
          num_x is None
```

```
Out[102]: True                Boolean type
```

```
In [103]: num_x = 10  
          num_x is None
```

```
Out[103]: False               Boolean type
```

# Type Casting

You can change the data type of a number using type casting.

```
In [58]: float_number = 3.6467
```

Float number

```
In [59]: float_number
```

```
Out[59]: 3.6467
```

```
In [60]: int(float_number)
```

Type cast to integer

```
Out[60]: 3
```

```
In [61]: str(float_number)
```

Type cast to string value

```
Out[61]: '3.6467'
```

# Data Structure: Tuple

A tuple is a one-dimensional, immutable ordered sequence of items which can be of mixed data types.

In [145]: `first_tuple = (12, 'Jack', 45.6, 'new', (3, 2), 'test')` Create a tuple

In [146]: `first_tuple`

Out[146]: `(12, 'Jack', 45.6, 'new', (3, 2), 'test')` View tuple

In [147]: `first_tuple[1]` Access the data at index value 1

Out[147]: `'Jack'`

In [148]: `first_tuple[1] = 'Mark'` Try to modify the tuple

`TypeError`

`<ipython-input-148-38afcbb40e37> in <module>()`  
----> 1 `first_tuple[1] = 'Mark'`

Traceback (most recent call last)

`TypeError: 'tuple' object does not support item assignment`

Error: A tuple is immutable and can't be modified

# Data Structure: Accessing Tuples

You can access a tuple using indices.

```
In [1]: first_tuple = (12, 'Jack', 45.6, 'new', (3,2), 'test') ← Tuple
```

```
In [2]: #Accessing elements using a positive index  
#The index count starts from the left, with the first index being 0  
first_tuple[2]
```

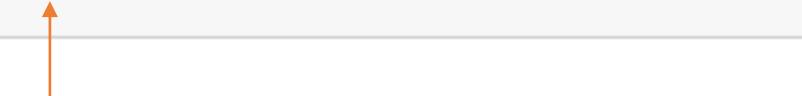
```
Out[2]: 45.6
```



Access with positive index

```
In [3]: #Accessing elements using a negative index  
#The index count starts from the right, with the first index being -1  
first_tuple[-3]
```

```
Out[3]: 'new'
```



Access with negative index

# Data Structure: Slicing Tuples

You can also slice a range of elements by specifying the start and end indices of the desired range.

```
In [1]: first_tuple = (12, 'Jack', 45.6, 'new', (3,2), 'test') ← Tuple
```

```
In [4]: #Creating a subset/slice of the tuple  
#Specify the indices of the elements, separated by a colon  
#The first index is inclusive; the second index is exclusive  
first_tuple[1:4] ←
```

Count starts with the first index  
but stops before the second index

```
Out [4]: ('Jack', 45.6, 'new')
```

```
In [5]: #You can use negative indices as well to slice a tuple  
#Count from the right, starting from -1, to specify the correct index  
first_tuple[1: -1] ←
```

```
Out [5]: ('Jack', 45.6, 'new', (3, 2))
```

Even for negative indices, the count  
stops before the second index

# Data Structure: List

A list is a one-dimensional, mutable ordered sequence of items which can be of mixed data types.

In [161]: `first_list = ['Mark', 101, 23.6, 'test', None, 11]` Create a list

In [162]: `first_list` View a list

Out[162]: `['Mark', 101, 23.6, 'test', None, 11]`

In [163]: `first_list.append('Jack')` Modify a list: Add new items  
`first_list`

Out[163]: `['Mark', 101, 23.6, 'test', None, 11, 'Jack']`

In [164]: `first_list.remove('Mark')` Modify a list: Remove items  
`first_list`

Out[164]: `[101, 23.6, 'test', None, 11, 'Jack']`

In [165]: `first_list.pop(2)` Access and remove list data using element indices

Out[165]: `'test'`

In [166]: `first_list.insert(1, 'Smith')` Modify a list: Insert a new item at a certain index  
`first_list`

Out[166]: `[101, 'Smith', 23.6, None, 11, 'Jack']`

# Data Structure: Accessing Lists

Just like tuples, you can access elements in a list through indices.

```
In [5]: first_list
```

```
Out[5]: [101, 'Smith', 'Smith', 23.6, None, 11, 'Jack'] ← New modified list
```

```
In [6]: #Accessing elements using a positive index  
#The index count starts from the left, with the first index being 0  
first_list[2]
```

```
Out[6]: 'Smith' ↑ Access with positive index
```

```
In [7]: #Accessing elements using a negative index  
#The index count starts from the right, with the first index being -1  
first_list[-2]
```

```
Out[7]: 11 ↑ Access with negative index
```

# Data Structure: Slicing Lists

Just like tuples, you can access elements in a list through indices.

```
In [5]: first_list
```

```
Out[5]: [101, 'Smith', 'Smith', 23.6, None, 11, 'Jack'] ← New modified list
```

```
In [8]: #Creating a subset/slice of the tuple  
#Specify the indices of the elements, separated by a colon  
#The first index is inclusive; the second index is exclusive  
first_list[1:4] ←
```

Count starts with the first index  
but stops before the second index

```
Out[8]: ['Smith', 'Smith', 23.6]
```

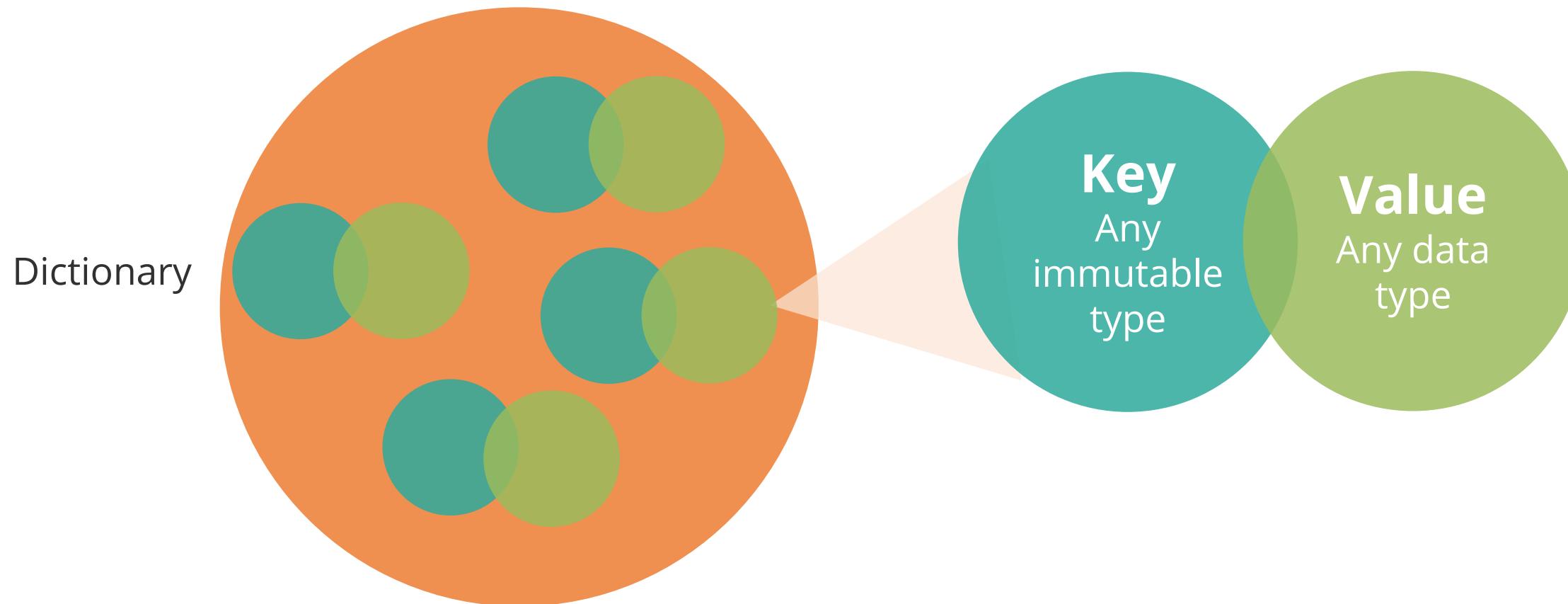
```
In [9]: #You can use negative indices as well to slice a tuple  
#Count from the right, starting from -1, to specify the correct index  
first_list[1:-1] ←
```

Even for negative indices, the count  
stops before the second index

```
Out[9]: ['Smith', 'Smith', 23.6, None, 11]
```

# Data Structure: Dictionary (dict)

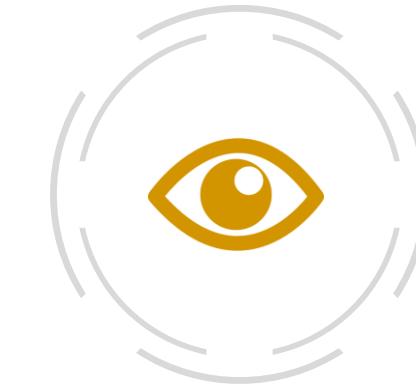
Dictionaries store a mapping between a set of keys and a set of values.



Define



Modify



View



Lookup

# Data Structure: View Dictionaries

You can view the keys and values in a dict, either separately or together, using the syntax shown here.

```
In [215]: first_dict = {'John':'john@abc.com','Kelly':'kelly@xyz.org','id':[23,81]}
```

Create a dictionary

```
In [216]: first_dict
```

View entire dictionary

```
Out[216]: {'John': 'john@abc.com', 'Kelly': 'kelly@xyz.org', 'id': [23, 81]}
```

```
In [217]: first_dict.keys()
```

View only keys

```
Out[217]: ['Kelly', 'John', 'id']
```

```
In [218]: first_dict.values()
```

View only values

```
Out[218]: ['kelly@xyz.org', 'john@abc.com', [23, 81]]
```

# Data Structure: Access and Modify dict Elements

You can also access and modify individual elements in a dict.

```
In [219]: first_dict['Kelly']
```

```
Out[219]: 'kelly@xyz.org'
```

```
In [220]: first_dict['id']
```

```
Out[220]: [23, 81]
```

```
In [221]: first_dict.update({'id':[32,55]})
```

```
In [222]: first_dict
```

```
Out[222]: {'John': 'john@abc.com', 'Kelly': 'kelly@xyz.org', 'id': [32, 55]}
```

```
In [223]: del first_dict['id']
```

```
In [224]: first_dict
```

```
Out[224]: {'John': 'john@abc.com', 'Kelly': 'kelly@xyz.org'}
```

Access with key

Modify dictionary:  
update

Modify dictionary:  
delete

# Data Structure: Set

A set is an unordered collection of unique elements.

```
In [327]: auto_survey = set(['Audi','BMW','BMW','Ferrari','GM','Mercedes','Cheverolet','GM'])
```

Create a set

```
In [328]: auto_survey
```

View the set

```
Out[328]: {'Audi', 'BMW', 'Cheverolet', 'Ferrari', 'GM', 'Mercedes'}
```

```
In [329]: auto_survey_set = {'Audi','BMW','BMW','Ferrari','GM','Mercedes','Cheverolet','GM'}
```

Create a set

```
In [330]: type(auto_survey_set)
```

View the object type

```
Out[330]: set
```

```
In [331]: auto_survery_set
```

View the set

```
Out[331]: {'Audi', 'BMW', 'Cheverolet', 'Ferrari', 'GM', 'Mercedes'}
```

# Data Structure: Set Operations

Let us look at some basic set operations.

```
In [334]: auto_survery_1 = set(['Audi', 'BMW', 'BMW', 'Ferrari', 'GM', 'Mercedes', 'Cheverolet', 'GM', 'Toyota'])  
auto_survery_2 = set(['BMW', 'Ferrari', 'GM', 'Hyundai', 'Kia', 'Cheverolet', 'GM', 'Ford', 'Toyota', 'Zen'])
```

Create sets

```
In [335]: combined_survery_report = auto_survery_1 | auto_survery_2
```

OR – Union  
set operation

```
In [336]: combined_survery_report
```

```
Out[336]: {'Audi',  
           'BMW',  
           'Cheverolet',  
           'Ferrari',  
           'Ford',  
           'GM',  
           'Hyundai',  
           'Kia',  
           'Mercedes',  
           'Toyota',  
           'Zen'}
```

View the output of the OR  
operation

```
In [337]: common_survey_report = auto_survery_1 & auto_survery_2
```

AND – Intersection set operation

```
In [338]: common_survey_report
```

```
Out[338]: {'BMW', 'Cheverolet', 'Ferrari', 'GM', 'Toyota'}
```

View the output of the  
NOT operation

# Basic Operator: “in”

The “in” operator is used to generate a Boolean value to indicate whether a given value is present in the container or not.

```
In [225]: student_list = ['Tom', 'Jack', 'Nick', 'Sarah', 'Nicole'] ← Create a list
```

```
In [226]: 'Nick' in student_list
```

```
Out[226]: True
```

```
In [227]: 'Mark' in student_list
```

```
Out[227]: False
```

Test presence of string with ‘in’ operator

```
In [228]: word = 'encyclopedia' ← Create a string
```

```
In [229]: 't' in word
```

```
Out[229]: False
```

```
In [230]: 'i' in word
```

```
Out[230]: True
```

Test presence of substrings with ‘in’ operator

## Basic Operator: “+”

The “plus” operator produces a new tuple, list, or string whose value is the concatenation of its arguments.

```
In [239]: test_score_1 = (68,96,71)  
test_score_2 = (92,87,83)
```

} Create tuples

```
In [240]: test_score = test_score_1+test_score_2  
test_score
```

} Add tuples

```
Out[240]: (68, 96, 71, 92, 87, 83)
```

```
In [241]: country_list_1 = ['USA','UK','China','Brazil','Mexico']  
country_list_2 = ['Australia','Spain','Italy']
```

} Create lists

```
In [242]: country_list_final = country_list_1+country_list_2  
country_list_final
```

} Add lists

```
Out[242]: ['USA', 'UK', 'China', 'Brazil', 'Mexico', 'Australia', 'Spain', 'Italy']
```

```
In [243]: first_name = 'George'  
last_name = 'Washington'
```

} Create strings

```
In [244]: full_name = first_name+' '+ last_name  
full_name
```

} Concatenate strings

```
Out[244]: 'George Washington'
```

## Basic Operator: “\*”

The “multiplication” operator produces a new tuple, list, or string that “repeats” the original content.

```
In [249]: age = (12,17,9) * 3 ← * operator with tuple  
age
```

```
Out[249]: (12, 17, 9, 12, 17, 9, 12, 17, 9)
```

```
In [250]: ID = [101,23,77,45] * 2 ← * operator with list  
ID
```

```
Out[250]: [101, 23, 77, 45, 101, 23, 77, 45]
```

```
In [251]: name = 'friend'*3 ← * operator with string  
name
```

```
Out[251]: 'friendfriendfriend'
```



The “\*” operator does not actually multiply the values; it only repeats the values for the specified number of times.

# Functions

Functions are the primary method of code organization and reuse in Python.

## Syntax

```
def <name>(arg1, arg2, ..., argN):  
    <statements>  
    return <value>
```

## Properties

- Outcome of the function is communicated by return statement
- Arguments in parenthesis are basically assignments

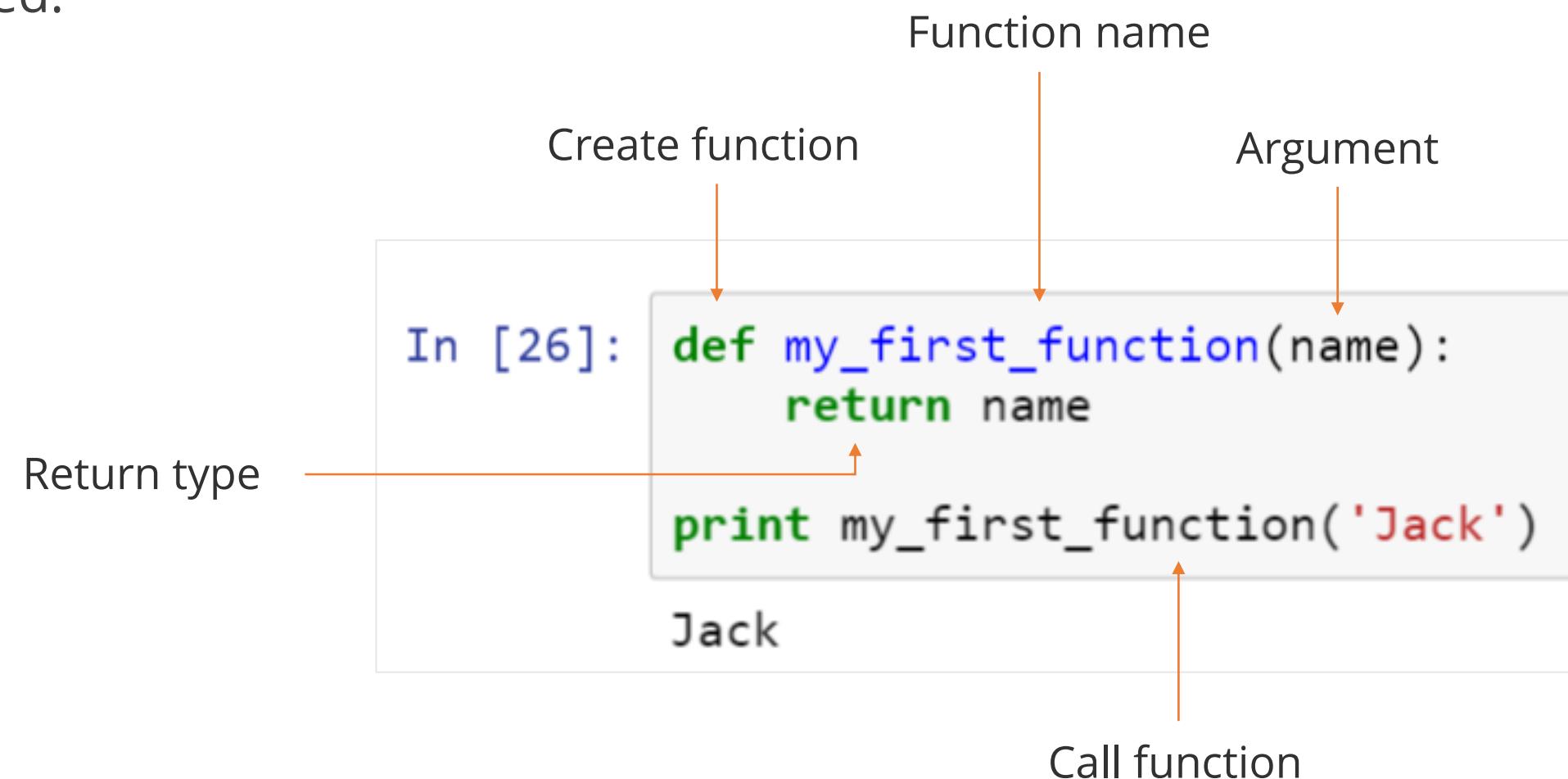


Use `def` to create a function and assign it a name.

# Functions: Considerations

Some important points to consider while defining functions:

- A function should always have a “return” value.
- If “return” is not defined, then it returns “None.”
- Function overloading is not permitted.



# Functions: Returning Values

You can use a function to return a single value or multiple values.

```
In [256]: def add_two_numbers(num1, num2): ← Create function
          return num1+num2
```

```
number1 = 23
number2 = 47.5
result = add_two_numbers(number1,number2) ← Call function
result
```

Out[256]: 70.5

```
In [257]: def profile(): ← Create function
```

```
    age = 21
    height = 5.5
    weight = 130
    return age, height, weight ← Multiple return
```

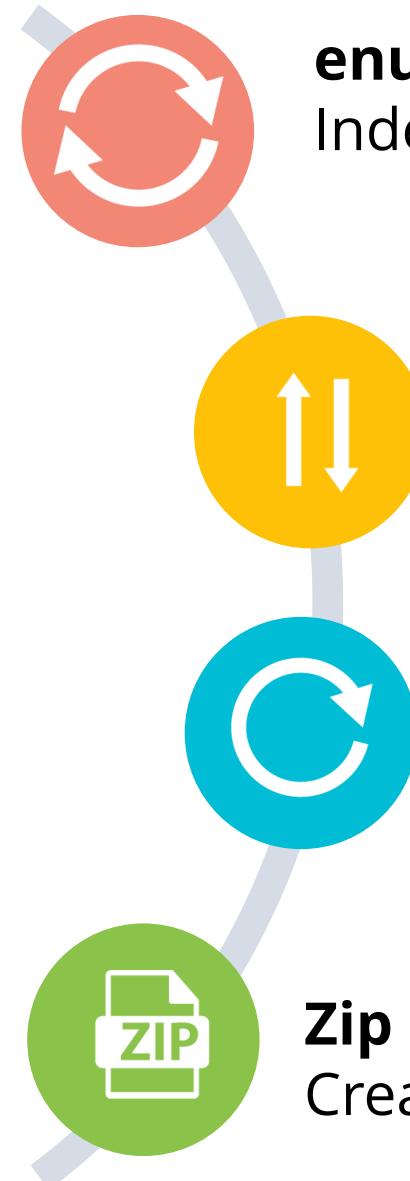
```
age, height, weight = profile() ← Call function
```

```
In [258]: print age, height, weight
```

21 5.5 130

# Built-in Sequence Functions

The built-in sequence functions of Python are as follows:



## enumerate

Indexes data to keep track of indices and corresponding data mapping

## sorted

Returns the new sorted list for the given sequence

## reversed

Iterates the data in reverse order

## Zip

Creates lists of tuples by pairing up elements of lists, tuples, or other sequence

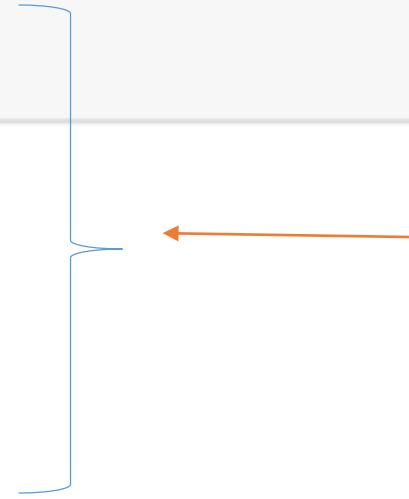
# Built-in Sequence Functions: enumerate

```
In [20]: store_list = ['McDonald', 'Taco Bell', 'Dunkin', 'Wendys', 'Chipotle']
```

List of food stores

```
In [21]: for position, name in enumerate(store_list):  
    print position, name
```

```
0 McDonald  
1 Taco Bell  
2 Dunkin  
3 Wendys  
4 Chipotle
```



Print data element and index using enumerate method

```
In [22]: store_map = dict((name, position) for position, name in enumerate(store_list))
```



Create a data element and index map using dict

```
In [23]: store_map
```

```
Out[23]: {'Chipotle': 4, 'Dunkin': 2, 'McDonald': 0, 'Taco Bell': 1, 'Wendys': 3}
```



View the store map in the form of key-value pair

# Built-in Sequence Functions: sorted

This screen explains the `sorted` function

In [27]: `sorted([91,43,65,56,7,33,21])` ← Sort numbers

Out[27]: [7, 21, 33, 43, 56, 65, 91]

In [28]: `sorted('the data science')` ← Sort a string value

Out[28]: [' ',  
 ',  
 'a',  
 'a',  
 'c',  
 'c',  
 'd',  
 'e',  
 'e',  
 'e',  
 'h',  
 'i',  
 'n',  
 's',  
 't',  
 't']

# Built-in Sequence Functions: reversed and zip

Let us see how to use reversed and zip functions

In [50]: `num_list = range(15)`

Create a list of numbers for range 15

In [51]: `list(reversed(num_list))`

Use reversed function to reverse the order

Out[51]: `[14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]`

In [52]: `subjects = ['math', 'statistics', 'algebra']  
subject_count = ['one', 'two', 'three']`

Define list of subjects and count

In [53]: `total_subject = zip(subjects, subject_count)  
total_subject`

Zip function to pair the data elements of lists

Out[53]: `[('math', 'one'), ('statistics', 'two'), ('algebra', 'three')]`

Returns list of tuples

In [54]: `type(total_subject)`

View type

Out[54]: `list`

# Control Flow: if, elif, else

The “if”, “elif,” and “else” statements are the most commonly used control flow statements.

```
In [341]: age = 21
```

If condition

```
In [342]: if age<18:  
    print 'minor'  
else:  
    print 'adult'
```

Else block

adult

```
In [343]: marks = 81
```

```
In [344]: if marks>90:  
    print 'grade A'  
elif 80<=marks<=90:  
    print 'grade B'  
elif 70<=marks<=80:  
    print 'grade C'  
elif 60<=marks<=70:  
    print 'grade D'  
else:  
    print 'grade F'
```

Nested if, elif and else

grade B

# Control Flow: “for” Loops

A “for” loop is used to iterate over a collection (like a list or tuple) or an iterator.

```
In [278]: stock_tickers =['AAPL','MSFT','GOOGL',None,'AMZN','CSCO','ORCL']
```

```
In [279]: for tickers in (stock_tickers):
    if(tickers is None):
        continue
    print tickers
```

For loop iterator

The ‘continue’ statement

```
AAPL  
MSFT  
GOOGL  
AMZN  
CSCO  
ORCL
```

```
In [280]: for tickers in (stock_tickers):
    if(tickers is None):
        break
    print tickers
```

The ‘break’ statement

```
AAPL  
MSFT  
GOOGL
```

# Control Flow: “while” Loops

A while loop specifies a condition and a block of code that is to be executed until the condition evaluates to False or the loop is explicitly ended with break.

In [283]: `temperature = 100  
while temperature > 95:  
 print(temperature)  
 temperature = temperature - 1`

While condition

```
100  
99  
98  
97  
96
```

# Control Flow: Exception Handling

Handling Python errors or exceptions gracefully is an important part of building robust programs and algorithms.

```
In [307]: def test_float(number):
           return float(number)
```

Create function

```
In [308]: test_float(7.32453)
```

```
Out[308]: 7.32453
```

```
In [309]: test_float('test float')
```

Pass wrong argument type

```
-----  
ValueError                                Traceback (most recent call last)
<ipython-input-309-d3d4bead5fb> in <module>()
----> 1 test_float('test float')

<ipython-input-307-c9efb2931c9f> in test_float(number)
      1 def test_float(number):
      2     return float(number)
```

Error

```
ValueError: could not convert string to float: test float
```

```
In [310]: def test_float(number):
           try:
               return float(number)
           except ValueError:
               return 'not a number, the input value is',number
```

Exception handling with try -except block

```
In [311]: test_float('test')
```

```
Out[311]: ('not a number, the input value is', 'test')
```



**QUIZ****1**

**What is the data type of the object  $x = 3 * 7.5$ ?**

- a. Int
- b. Float
- c. String
- d. None of the above



**QUIZ****1**

**What is the data type of the object  $x = 3 * 7.5$ ?**

- a. Int
- b. Float
- c. String
- d. None of the above



The correct answer is **b**.

**Explanation:** Since one of the operands is float, the  $x$  variable will also be of the float data type.

**QUIZ**  
**2**

**Which of the data structures can be modified? *Select all that apply.***

- a. tuple
- b. list
- c. dict
- d. set



**QUIZ**  
**2**

**Which of the data structures can be modified? *Select all that apply.***

- a. tuple
- b. list
- c. dict
- d. set



The correct answer is **b, c, d**

**Explanation:** Only a tuple is immutable and cannot be modified. All the other data structures can be modified.

# QUIZ

3

## What will be the output of the following code?

```
In [350]: summit_venue = ['NYC', 'LA', 'Miami', 'London', 'Madrid', 'Paris']
summit_venue[3:-1]
```

- a. ['NYC', 'Madrid']
- b. ['London', 'Madrid']
- c. ['Miami', 'Madrid']
- d. ['Miami', 'Paris']



## QUIZ

3

### What will be the output of the following code?

```
In [350]: summit_venue = ['NYC', 'LA', 'Miami', 'London', 'Madrid', 'Paris']
summit_venue[3:-1]
```

- a. ['NYC', 'Madrid']
- b. ['London', 'Madrid']
- c. ['Miami', 'Madrid']
- d. ['Miami', 'Paris']



The correct answer is **b**.

**Explanation:** Slicing starts at the first index and stops before the second index. Here, the element at index 3 is "London" and the element before index -1 is "Madrid."

**QUIZ**  
**4**

**Which of the following data structures is preferred to contain a unique collection of values?**

- a. dict
- b. list
- c. set
- d. tuple



**QUIZ**  
**4**

**Which of the following data structures is preferred to contain a unique collection of values?**

- a. dict
- b. list
- c. set
- d. tuple

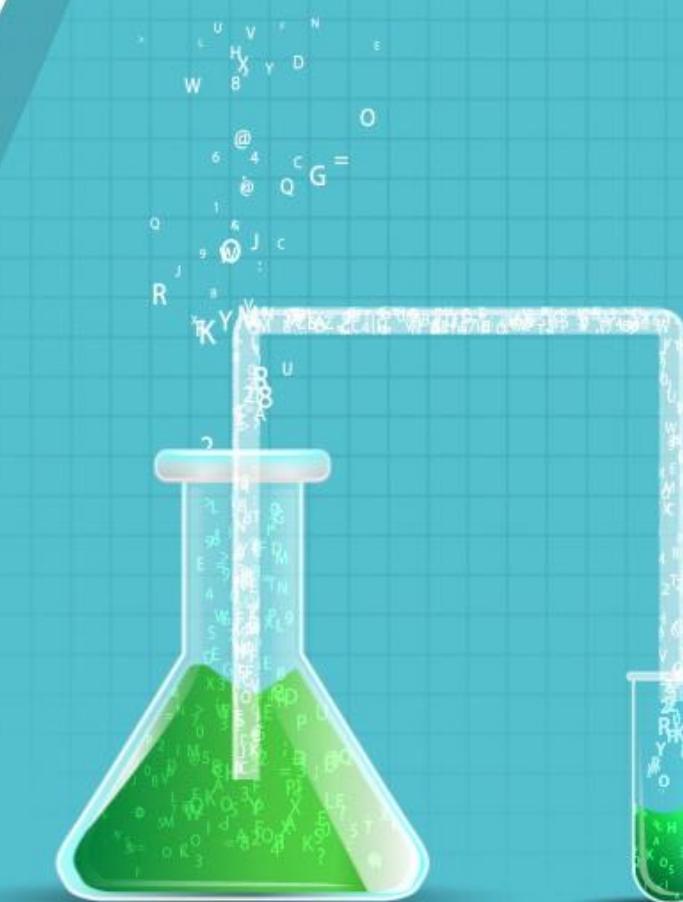


The correct answer is **c**.

**Explanation:** A set is used when a unique collection of values is desired.

# Key Takeaways

- Download Python 2.7 version from Anaconda and install Jupyter notebook.
- When you assign values to variables, you create references and not duplicates.
- Integers, floats, strings, None, and Boolean are some of the data types supported by Python.
- Tuples, lists, dicts, and sets are some of the data structures of Python.
- You can use indices to access individual or a range of elements in a data structure.
- The “in”, “+”, and “\*” are some of the basic operators.
- Functions are the primary and the most important methods of code organization and reuse in Python.
- The conditional “if”, “elif” statements, “while” and “for” loops, and exception handling are some important control flow statements.



**This concludes “Python: Environment Setup and Essentials.”**  
The next lesson is “Mathematical Computing with Python (NumPy).”

DATA  
SCIENCE

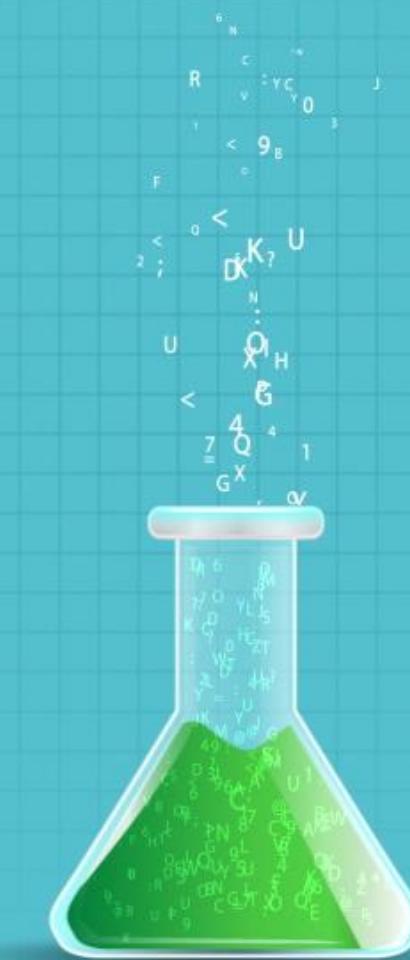
# Data Science with Python

## Lesson 5—Mathematical Computing with Python (NumPy)



# What You'll Learn

- What NumPy is and why it is important
- Basics of NumPy, including its fundamental objects
- Create and print a NumPy array
- Carry out basic operations in NumPy
- Use shape manipulation and copying methods
- Execute linear algebraic functions
- Build basic programs using NumPy



# Quick Recap: Lists

A list is a collection of values. You can individually add, remove, or update these values. A single list can contain multiple data types.

List

```
distance=[10,15,17,26]
```

Collection of values

```
time=[.30,.47,.55,1.20]
```

Multiple types (heterogeneous)

Add, remove, update

# Limitations of Lists

Though you can change individual values in a list, you cannot apply a mathematical operation over the entire list.

```
distance=[10,15,17,26]
time=[.30,.47,.55,1.20]
```

```
speed=distance/time
```

Mathematical operation over the entire “distance” and “time” lists

```
TypeError
<ipython-input-37-b779bad68500> in <module>()
----> 1 speed=distance/time
```

Traceback (most recent call last)

```
TypeError: unsupported operand type(s) for /: 'list' and 'list'
```

Error

# Why NumPy

Numerical Python (NumPy) supports multidimensional arrays over which you can easily apply mathematical operations.

```
distance=[10,15,17,26]  
time=[.30,.47,.55,1.20]
```

```
import numpy as np
```

Import NumPy

```
np_distance = np.array(distance)  
np_time=np.array(time)  
speed=np_distance/np_time
```

Create “distance” and “time” NumPy arrays

Mathematical function applied over the entire “distance” and “time” arrays

```
speed
```

```
array([ 33.33333333, 31.91489362, 30.90909091, 21.66666667])
```

Output

# NumPy Overview

NumPy is the foundational package for mathematical computing in Python.

It has the following properties:

Supports fast and efficient multidimensional arrays (ndarray)



Performs linear algebraic operations, Fourier transforms, and random number generation



Efficient way of storing and manipulating data



Executes element-wise computations and mathematical calculations



Tools for reading/writing array based datasets to disk

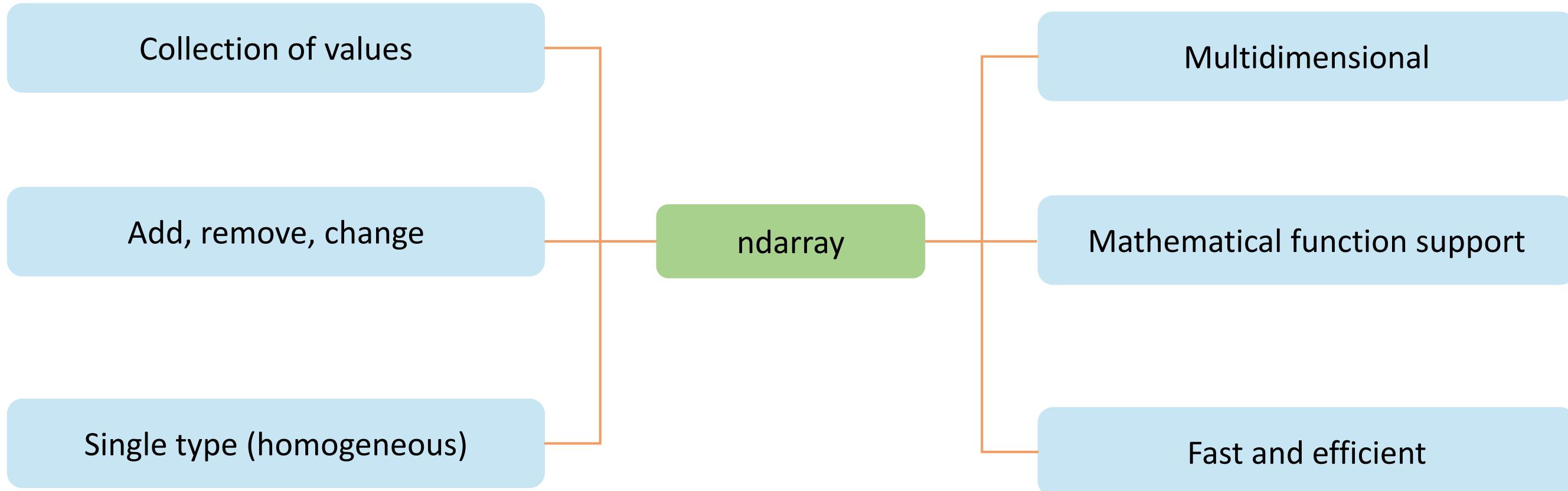


Tools for integrating language codes (C, C++)



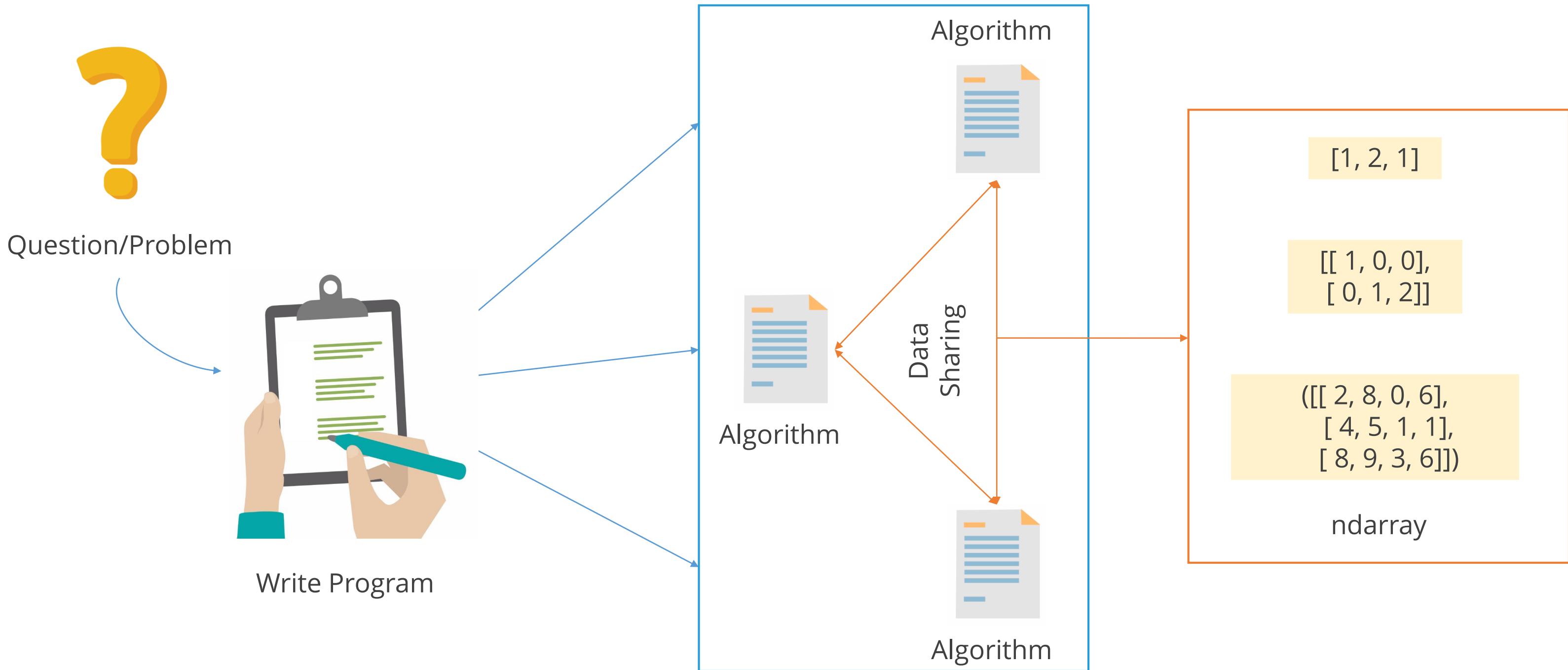
# Properties of ndarray

An array in NumPy has the following properties:



# Purpose of ndarray

The ndarray in Python is used as the primary container to exchange data between algorithms.



# Knowledge Check—Sequence it Right!

The code here is buggy. You have to correct its sequence to debug it.

1

```
distance=[10,15,17,26]
time=[.30,.47,.55,1.20]
```

2

```
np_distance = np.array(distance)
np_time=np.array(time)
```

3

```
import numpy as np
```

4

```
speed=np_distance/np_time
speed
array([ 33.33333333,  31.91489362,  30.90909091,  21.66666667])
```

# Knowledge Check—Sequence it Right!

The code here is buggy. You have to correct its sequence to debug it.

1

```
distance=[10,15,17,26]
time=[.30,.47,.55,1.20]
```

2

```
import numpy as np
```

3

```
np_distance = np.array(distance)
np_time=np.array(time)
```

4

```
speed=np_distance/np_time
```

```
speed
```

```
array([ 33.33333333,  31.91489362,  30.90909091,  21.66666667])
```

# Types of Arrays

Arrays can be one-dimensional, two dimensional, three-dimensional, or multi-dimensional.

## One-Dimensional Array

Printed as rows

array([5, 7, 9]) ←  
Length = 3

5	7	9
0	1	2
<i>x axis</i>		

## Two-Dimensional Array

Printed as matrices (2x3)

array([[ 0, 1, 2],  
[ 5, 6, 7]])

Length = 3

y axis		0 (0,0)	1 (0,1)	2 (0,2)
x axis		5 (1,0)	6 (1,1)	7 (1,2)

## Three-Dimensional Array

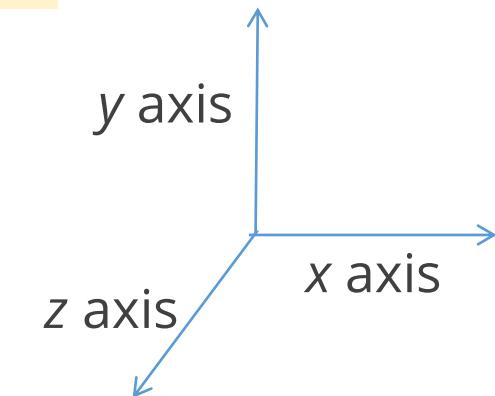
Printed as list of matrices (3x3x3)

array([[[ 0, 1, 2],  
[ 3, 4, 5],  
[ 6, 7, 8]],

[[ 9, 10, 11],  
[12, 13, 14],  
[15, 16, 17]],

[[18, 19, 20],  
[21, 22, 23],  
[24, 25, 26]]])

Length = 3



## Demo 01—Creating and Printing an ndarray

Demonstrate how to create and print an ndarray.



# Knowledge Check

## How many elements will the following code print?

```
print(np.linspace(4,13,7))
```

- a. 4
- b. 7
- c. 11
- d. 13



## How many elements will the following code print?

```
print(np.linspace(4,13,7))
```

- a. 4
- b. 7
- c. 11
- d. 13



The correct answer is **b**.

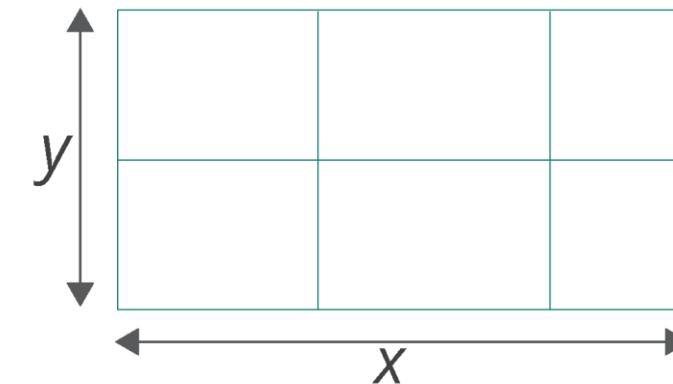
**Explanation:** In the “linspace” function, “4” is the starting element and “13” is the end element. The last number “7” specifies that a total of seven equally spaced elements should be created between “4” and “13,” both numbers inclusive. In this case, the “linspace” function creates the following array: [ 4. 5.5 7. 8.5 10. 11.5 13. ]

# Class and Attributes of ndarray—.ndim

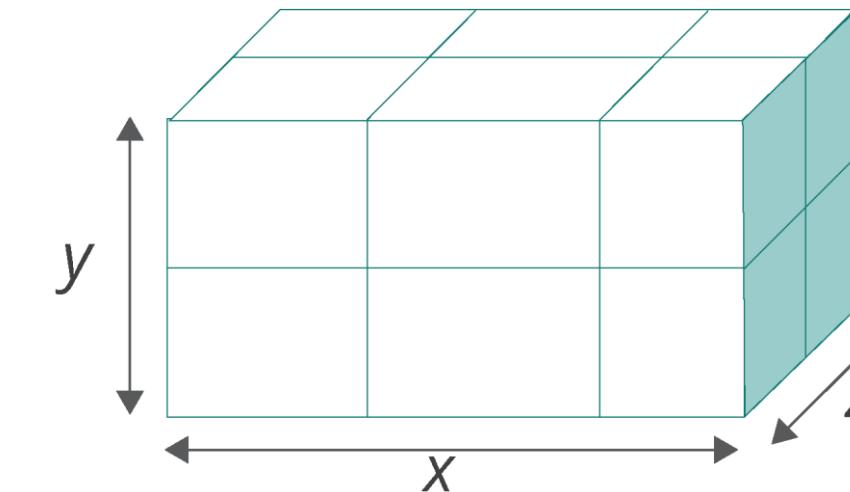
Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

This refers to the number of axes (dimensions) of the array. It is also called the rank of the array.



Two axes or 2D array



Three axes or 3D array

ndarray.size

ndarray.dtype

Concept

Example

# Class and Attributes of ndarray—.ndim

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

The array "np\_city" is one-dimensional, while the array "np\_city\_with\_state" is two-dimensional.

```
In [108]: np_city = np.array(['NYC', 'LA', 'Miami', 'Houston'])
```

```
In [109]: np_city.ndim
```

```
Out[109]: 1
```

```
In [110]: np_city_with_state = np.array([[ 'NYC', 'LA', 'Miami', 'Houston'], ['NY', 'CA', 'FL', 'TX']])
```

```
In [111]: np_city_with_state.ndim
```

```
Out[111]: 2
```

ndarray.size

ndarray.dtype

Concept

Example

# Class and Attributes of ndarray—.shape

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

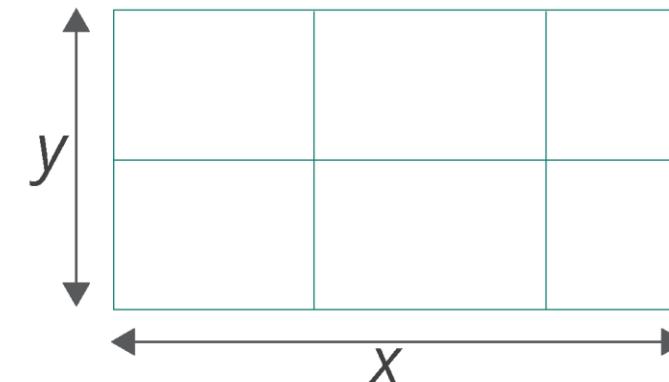
ndarray.ndim

ndarray.shape

ndarray.size

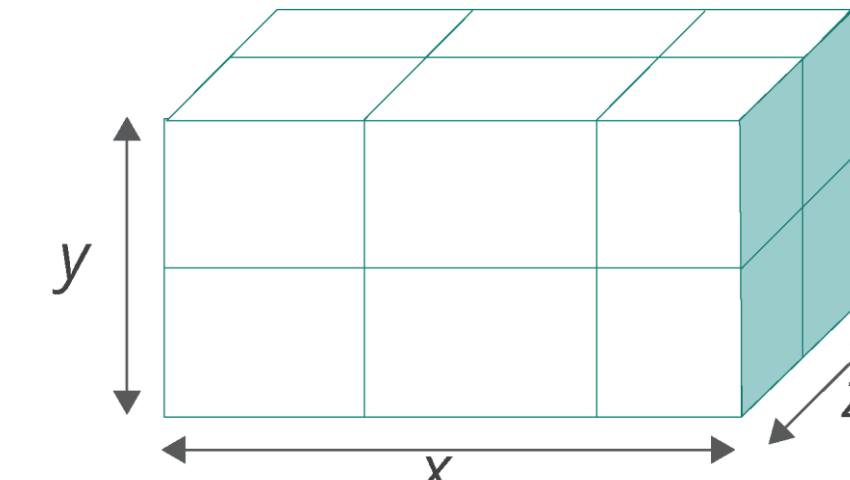
ndarray.dtype

This consists of a tuple of integers showing the size of the array in each dimension. The length of the "shape tuple" is the rank or ndim.



2 rows, 3 columns

Shape: (2, 3)



2 rows, 3 columns, 2 ranks

Shape: (2, 3, 2)

Concept

Example

# Class and Attributes of ndarray—.shape

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

The shape tuple of both the arrays indicate their size along each dimension.

ndarray.shape

```
In [108]: np_city = np.array(['NYC', 'LA', 'Miami', 'Houston'])
```

```
In [110]: np_city_with_state = np.array([[['NYC', 'LA', 'Miami', 'Houston'], ['NY', 'CA', 'FL', 'TX']]])
```

```
In [112]: np_city.shape
```

```
Out[112]: (4L,)
```

```
In [113]: np_city_with_state.shape
```

```
Out[113]: (2L, 4L)
```

ndarray.size

ndarray.dtype

Concept

Example

# Class and Attributes of ndarray—.size

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

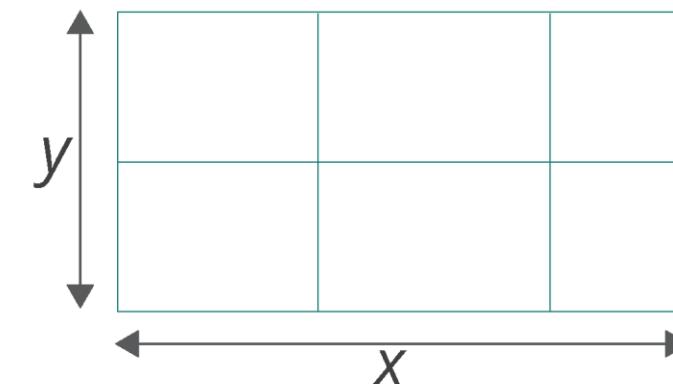
ndarray.ndim

ndarray.shape

ndarray.size

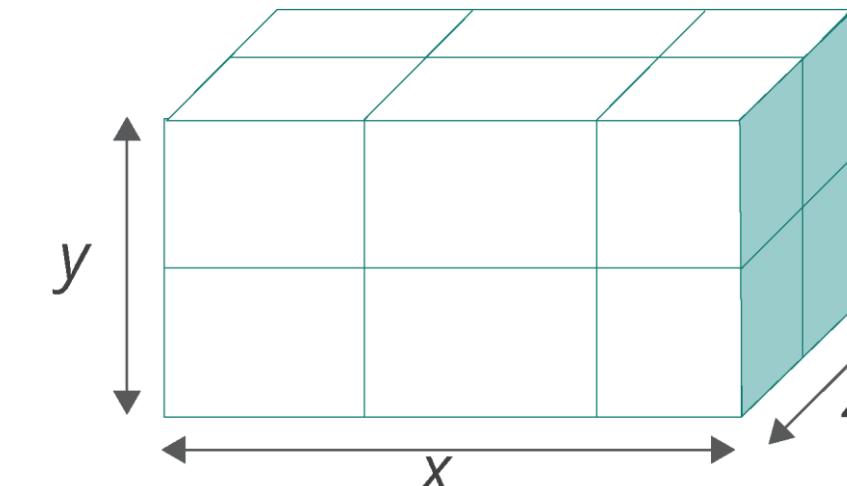
ndarray.dtype

It gives the total number of elements in the array. It is equal to the product of the elements of the shape tuple.



Array contains 6 elements

Array a = (2, 3)  
Size = 6



Array contains 12 elements

Array b = (2, 3, 2)  
Size = 12

Concept

Example

# Class and Attributes of ndarray—.size

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

Look at the examples to see how the shape tuples of the arrays are used to calculate their size.

In [112]: `np_city.shape`

Out[112]: `(4L,)`

In [113]: `np_city_with_state.shape`

Out[113]: `(2L, 4L)`

In [114]: `np_city.size`

Out[114]: `4`

In [115]: `np_city_with_state.size`

Out[115]: `8`

ndarray.size

ndarray.dtype

Concept

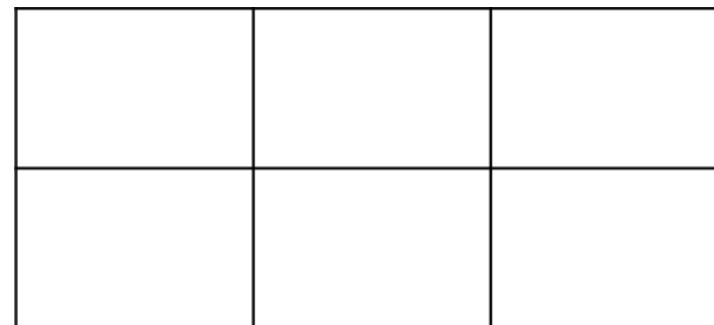
Example

# Class and Attributes of ndarray—.dtype

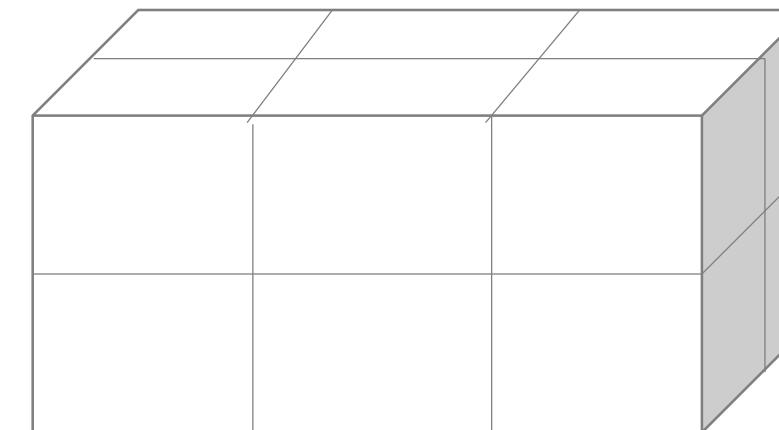
Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

It's an object that describes the type of the elements in the array. It can be created or specified using Python.



ndarray.shape



ndarray.size

Array contains integers

Array a = [3, 7, 4]  
[2, 1, 0]

Array contains floats

Array b = [1.3, 5.2, 6.7]  
[0.2, 8.1, 9.4]

[2.6, 4.2, 3.9]  
[7.8, 3.4, 0.8]

ndarray.dtype

Concept

Example

# Class and Attributes of ndarray—.dtype

Numpy's array class is "ndarray," also referred to as "numpy.ndarray." The attributes of ndarray are:

ndarray.ndim

Both the arrays are of "string" data type (dtype) and the longest string is of length 7, which is "Houston."

ndarray.shape

```
In [116]: np_city  
Out[116]: array(['NYC', 'LA', 'Miami', 'Houston'],  
                 dtype='|S7')
```

ndarray.size

```
In [117]: np_city_with_state  
Out[117]: array([['NYC', 'LA', 'Miami', 'Houston'],  
                  ['NY', 'CA', 'FL', 'TX']],  
                 dtype='|S7')
```

ndarray.dtype

```
In [118]: np_city_with_state.dtype  
Out[118]: dtype('S7')
```

Concept

Example

# Basic Operations

Using the following operands, you can easily apply various mathematical, logical, and comparison operations on an array.

## Mathematical Operations

Addition	+
Subtraction	-
Multiplication	*
Division	/
Exponentiation	**

## Logical Operations

And	&
Or	
Not	~

## Comparison Operations

Greater	>
Greater or equal	>=
Less	<
Less or equal	<=
Equal	==
Not equal	!=

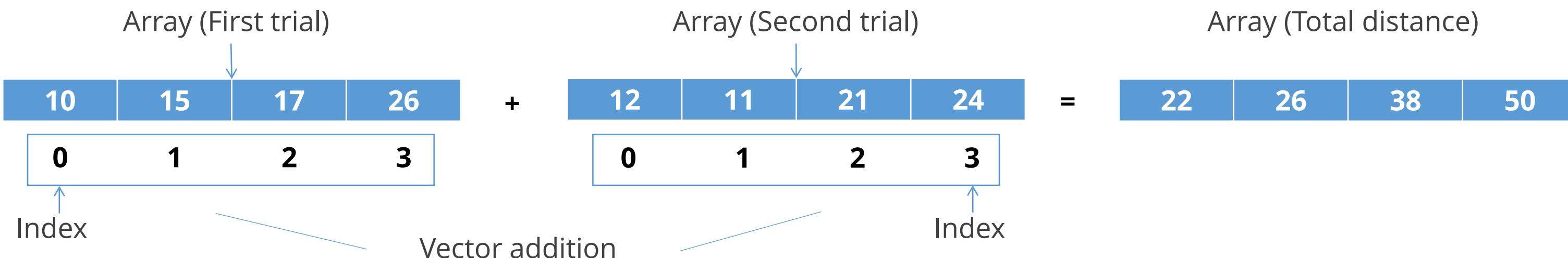
## Demo 03—Executing Basic Operations

Demonstrate how to apply some basic operations on an array.

# Basic Operations—Example

NumPy uses the indices of the elements in each array to carry out basic operations. In this case, where we are looking at a dataset of four cyclists during two trials, vector addition of the arrays gives the required output.

```
In [99]: first_trial_cyclist =[10,15,17,26]           First trial
In [100]: second_trial_cyclist =[12,11,21,24]          Second trial
In [101]: np_first_trial_cyclist = np.array(first_trial_cyclist)
In [102]: np_second_trial_cyclist = np.array(second_trial_cyclist)
In [103]: np_first_trial_cyclist+np_second_trial_cyclist   Total distance
Out[103]: array([22, 26, 38, 50])
```



# Accessing Array Elements: Indexing

You can access an entire row of an array by referencing its axis index.

1<sup>st</sup> set data      2nd set data



```
In [117]: cyclist_trials = np.array([[10,15,17,26],[12,11,21,24]]) ← Create 2D array using cyclist trial data shown earlier
```

```
In [118]: first_trial =cyclist_trials[0] ← First trial data
```

```
In [119]: first_trial
```

```
Out[119]: array([10, 15, 17, 26])
```

```
In [120]: second_trial = cyclist_trials[1] ← Second trial data
```

```
In [121]: second_trial
```

```
Out[121]: array([12, 11, 21, 24])
```

2D array containing cyclists' data

10	15	17	26
12	11	21	24

← First trial (axis 0)

← Second trial (axis 1)

# Accessing Array Elements: Indexing (contd.)

You can refer the indices of the elements in an array to access them. You can also select a particular index of more than one axis at a time.

```
In [122]: first_cyclist_firstTrial = cyclist_trials[0][0]
```

First cyclist: first trial data

```
In [123]: first_cyclist_firstTrial
```

```
Out[123]: 10
```

```
In [124]: first_cyclist_all_trials = cyclist_trials[:,0]
```

First cyclist: all trial data  
(Use ":" to select all the rows of an array)

```
In [125]: first_cyclist_all_trials
```

```
Out[125]: array([10, 12])
```

Cyclist 1, first trial data →

(0, 0)	(0, 1)	(0, 2)	(0, 3)
10	15	17	26
12	11	21	24
(1, 0)	(1, 1)	(1, 2)	(1, 3)

Cyclist 1, all trials data

(0, 0)	(0, 1)	(0, 2)	(0, 3)
10	15	17	26
12	11	21	24
(1, 0)	(1, 1)	(1, 2)	(1, 3)

# Accessing Array Elements: Slicing

Use the slicing method to access a range of values within an array.

Shape of the array

```
In [152]: cyclist_trials.shape
```

```
Out[152]: (2L, 4L)
```

```
In [153]: two_cyclist_trial_data=cyclist_trials[:,1:3]
```

```
In [154]: two_cyclist_trial_data
```

```
Out[154]: array([[15, 17],  
                  [11, 21]])
```

Slicing the array data `[ :, 1 : 3 ]`  
where 1 is inclusive but 3 is not

2 rows

Shape of the array

10	15	17	26
12	11	21	24

4 columns

Use ':' to select  
all rows

Slicing the array

10	15	17	26
12	11	21	24

0 1 2 3

Starting index (1)      Ending index (2)

# Activity—Slice It!

Select any two elements from the array to see how the statement required to slice the range changes.

## Rules of the Game

- Choose the first element of the range. Then, choose the element that ends the range.
- See how the values in the statement change according to your choices.
- Refresh to try again.

5	8	10	21
---	---	----	----

```
example_array[1:3]
```

Select any two elements from the array.

# Accessing Array Elements: Iteration

Use the iteration method to go through each data element present in the dataset.

```
In [117]: cyclist_trials = np.array([[10,15,17,26],[12,11,21,24]])
```

```
In [153]: two_cyclist_trial_data=cyclist_trials[:,1:3]
```

```
In [154]: two_cyclist_trial_data
```

```
Out[154]: array([[15, 17],  
                  [11, 21]])
```

```
In [159]: for iterate_cyclist_trials_data in cyclist_trials:  
          print (iterate_cyclist_trials_data)
```

```
[10 15 17 26]  
[12 11 21 24]
```

Iterate with “for loop”  
through entire dataset

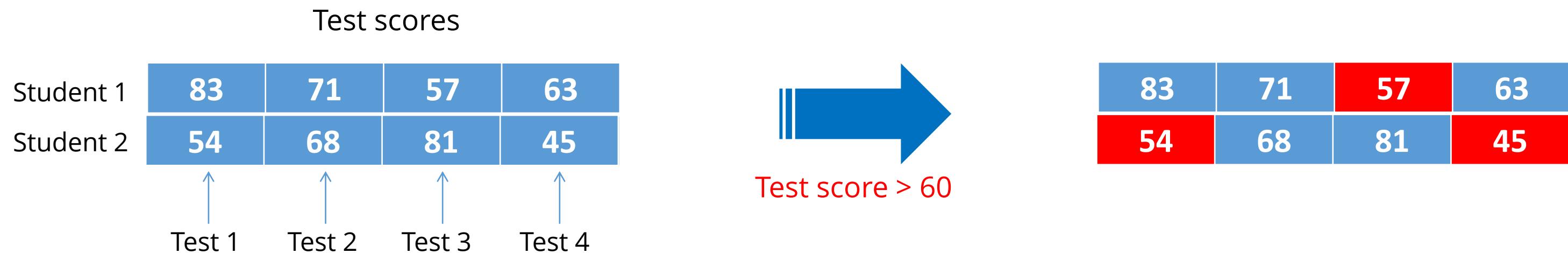
```
In [160]: for iterate_two_cyclist_trial_data in two_cyclist_trial_data:  
          print (iterate_two_cyclist_trial_data)
```

```
[15 17]  
[11 21]
```

Iterate with “for loop” through  
the “two cyclist” datasets

# Indexing with Boolean Arrays

Boolean arrays are useful when you need to select a dataset according to set criteria. Here, the original dataset contains test scores of two students. You can use a Boolean array to choose only the scores that are above a given value.



```
In [234]: test_scores = np.array([[83, 71, 57, 63], [54, 68, 81, 45]])
```

```
In [235]: passing_score = test_scores > 60 ← Setting the passing score
```

```
In [236]: passing_score
```

```
Out[236]: array([[ True,  True, False,  True],
   [False,  True,  True, False]], dtype=bool) ← Shows data elements which fit the
                                                criteria (Boolean array)
```

```
In [237]: test_scores[passing_score] ← Send "passing score" as an argument to "test scores" object
```

```
Out[237]: array([83, 71, 63, 68, 81])
```

# Copy and Views

When working with arrays, data is copied into new arrays only in some cases. Following are the three possible scenarios:



## Simple Assignments

In this method, a variable is directly assigned the value of another variable. No new copy is made.



## View/Shallow Copy

```
In [303]: NYC_Borough = np.array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Queens'])
```

```
In [294]: NYC_Borough
```

```
Out[294]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Queens'],
                 dtype='|S13')
```

← Original dataset

```
In [295]: Boroughs_in_NYC = NYC_Borough
```

```
In [296]: Boroughs_in_NYC
```

```
Out[296]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Queens'],
                 dtype='|S13')
```

← Assigned dataset

```
In [297]: Boroughs_in_NYC is NYC_Borough
```

```
Out[297]: True
```

Shows both objects are the same



## Deep Copy

# Copy and Views

When working with arrays, data is copied into new arrays only in some cases. There are three possible scenarios:



Simple Assignments

A view, also referred to as a shallow copy, creates a new array object.

```
In [296]: Boroughs_in_NYC  
Out[296]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Queens'],  
                 dtype='|S13')
```

Original dataset



View/Shallow Copy

```
In [298]: View_of_Borough_in_NYC = Boroughs_in_NYC.view()
```

```
In [299]: len(View_of_Borough_in_NYC)
```

```
Out[299]: 5
```

```
In [300]: View_of_Borough_in_NYC[4] = 'Central Park' ← Change value in "view" object
```



Deep Copy

```
In [301]: View_of_Borough_in_NYC
```

```
Out[301]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Central Park'],  
                 dtype='|S13')
```

```
In [302]: Boroughs_in_NYC
```

```
Out[302]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Central Park'],  
                 dtype='|S13')
```

Original dataset changed

# Copy and Views

When working with arrays, data is copied into new arrays only in some cases. There are three possible scenarios:



Simple Assignments



View/Shallow Copy



Deep Copy

Copy is also called “deep copy” because it entirely copies the original dataset. Any change in the copy will not affect the original dataset.

```
In [304]: Copy_of_NYC_Borough = NYC_Borough.copy()
```

Shows “copy” and original object are different

```
In [305]: Copy_of_NYC_Borough is NYC_Borough
```



```
Out[305]: False
```

```
In [306]: Copy_of_NYC_Borough.base is NYC_Borough
```



```
Out[306]: False
```

Shows “copy” object data is not owned by the original dataset

```
In [307]: Copy_of_NYC_Borough[4]='Central Park'
```

Change value in “copy”

```
In [308]: NYC_Borough
```

```
Out[308]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Queens'],  
                 dtype='|S13')
```



“Copy” object changed

```
In [309]: Copy_of_NYC_Borough
```

```
Out[309]: array(['Manhattan', 'Bronx', 'Brooklyn', 'Staten Island', 'Central Park'],  
                 dtype='|S13')
```



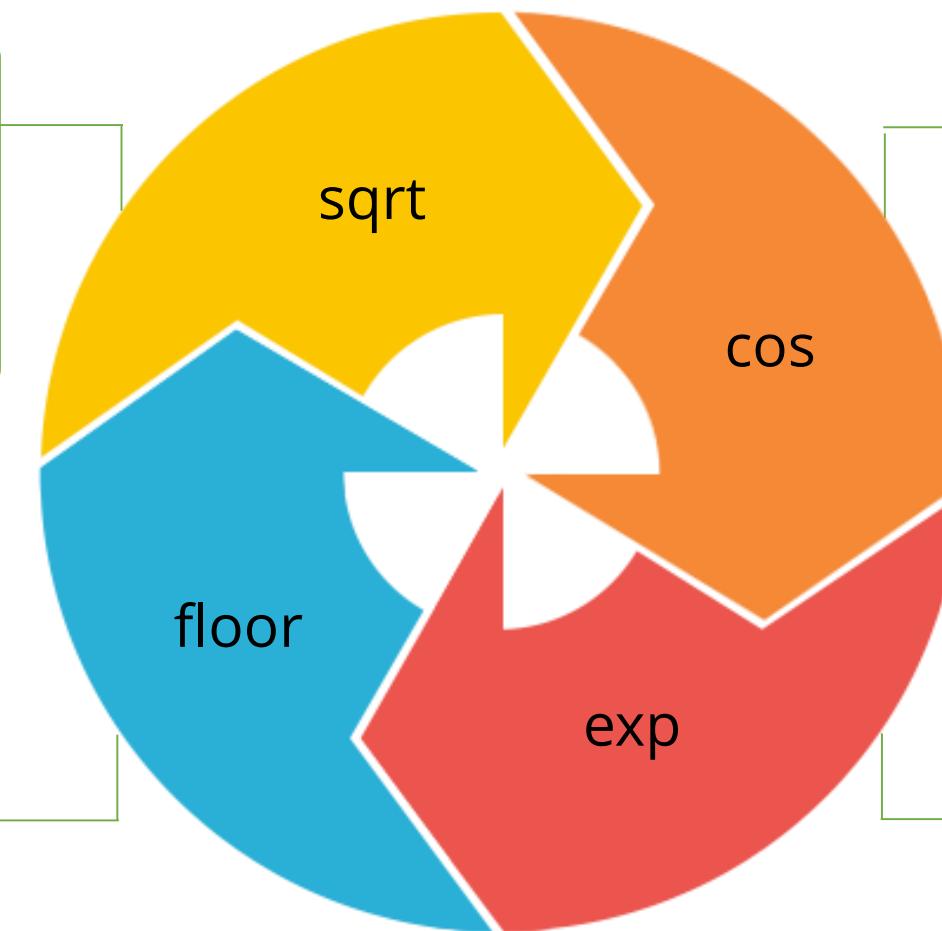
Original dataset retained

# Universal Functions (ufunc)

NumPy provides useful mathematical functions called Universal Functions. These functions operate element-wise on an array, producing another array as output. Some of these functions are listed here:

**sqrt** function provides the square root of every element in the array.

**floor** function returns the largest integer value of every element in the array.



**cos** function gives cosine values for all elements in the array.

**exp** function performs exponentiation on each element.

# ufunc—Examples

Let's look at some common ufunc examples:

```
In [186]: np_sqrt = np.sqrt([2,4,9,16])
```

Numbers for which square root will be calculated

```
In [187]: np_sqrt
```

```
Out[187]: array([ 1.41421356,  2.  ,  3.  ,  4.  ])
```

Square root values

```
In [188]: from numpy import pi
          np.cos(0)
```

Import pi\*

```
Out[188]: 1.0
```

```
In [189]: np.sin(pi/2)
```

Trigonometric functions

```
Out[189]: 1.0
```

```
In [190]: np.cos(pi)
```

```
Out[190]: -1.0
```

```
In [191]: np.floor([1.5,1.6,2.7,3.3,1.1,-0.3,-1.4])
```

Return the floor of the input element wise

```
Out[191]: array([ 1.,  1.,  2.,  3.,  1., -1., -2.])
```

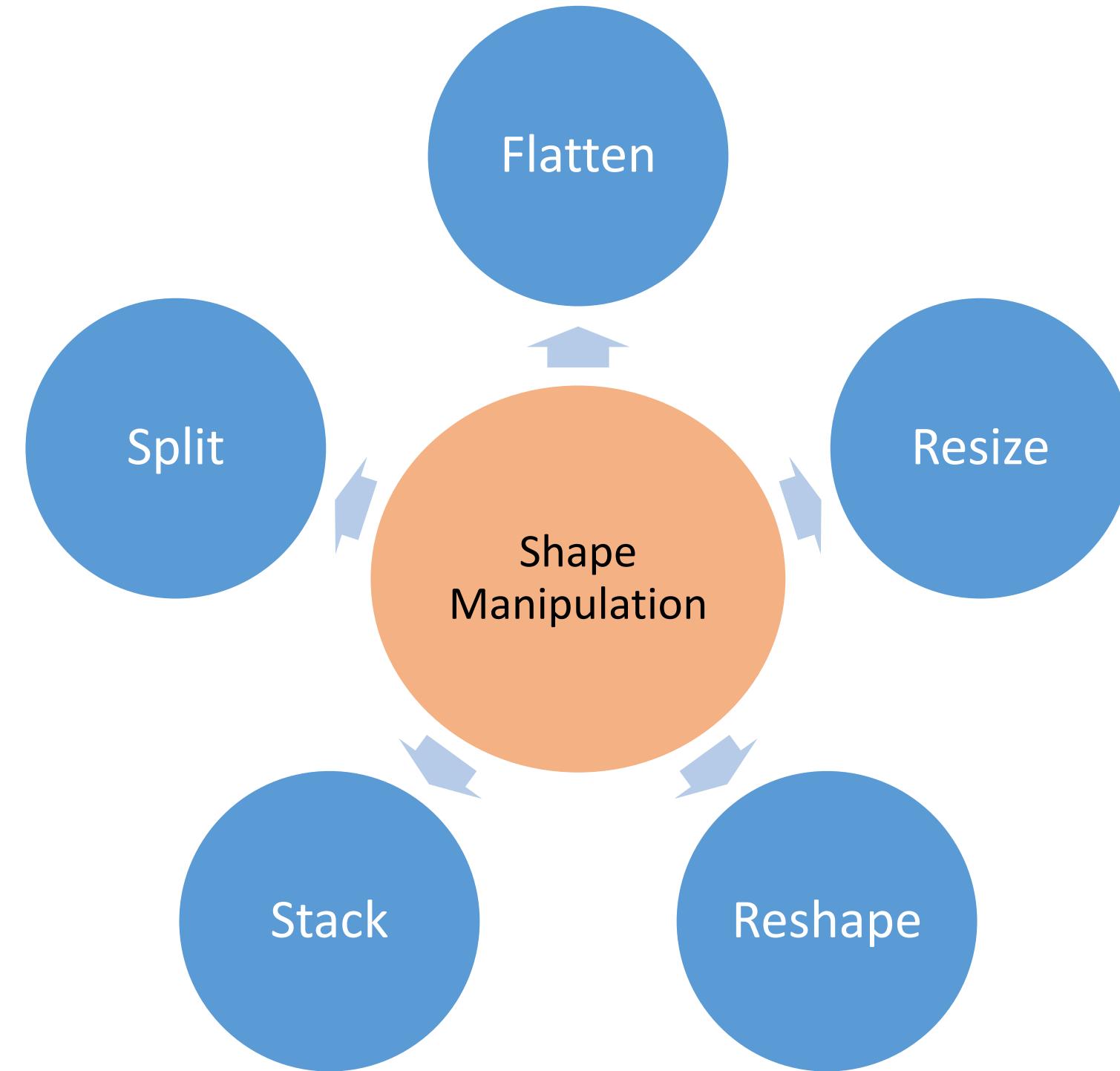
```
In [192]: np.exp([0,1,5])
```

```
Out[192]: array([ 1.        ,  2.71828183, 148.4131591 ])
```

Exponential functions for complex mathematical calculations

# Shape Manipulation

You can use certain functions to manipulate the shape of an array to do the following:



# Shape Manipulation—Example

You can use certain functions to manipulate the shape of an array to do the following:

```
In [383]: new_cyclist_trials = np.array([[10,15,17,26,13,19],[12,11,21,24,14,23]])
```

```
In [384]: new_cyclist_trials.ravel() ← Flattens the dataset
```

```
Out[384]: array([10, 15, 17, 26, 13, 19, 12, 11, 21, 24, 14, 23])
```

```
In [385]: new_cyclist_trials.reshape(3,4) ← Changes or reshapes the dataset to 3 rows and 4 columns
```

```
Out[385]: array([[10, 15, 17, 26],  
[13, 19, 12, 11],  
[21, 24, 14, 23]])
```

```
In [386]: new_cyclist_trials.resize(2,6) ← Resizes again to 2 rows and 6 columns
```

```
In [387]: new_cyclist_trials
```

```
Out[387]: array([[10, 15, 17, 26, 13, 19],  
[12, 11, 21, 24, 14, 23]])
```

```
In [388]: np.hsplit(new_cyclist_trials,2) ← Splits the array into two
```

```
Out[388]: [array([[10, 15, 17],  
[12, 11, 21]]), array([[26, 13, 19],  
[24, 14, 23]])]
```

```
In [389]: new_cyclist_1 = np.array([10,15,17,26,13,19])
```

```
In [390]: new_cyclist_2 = np.array([12,11,21,24,14,23])
```

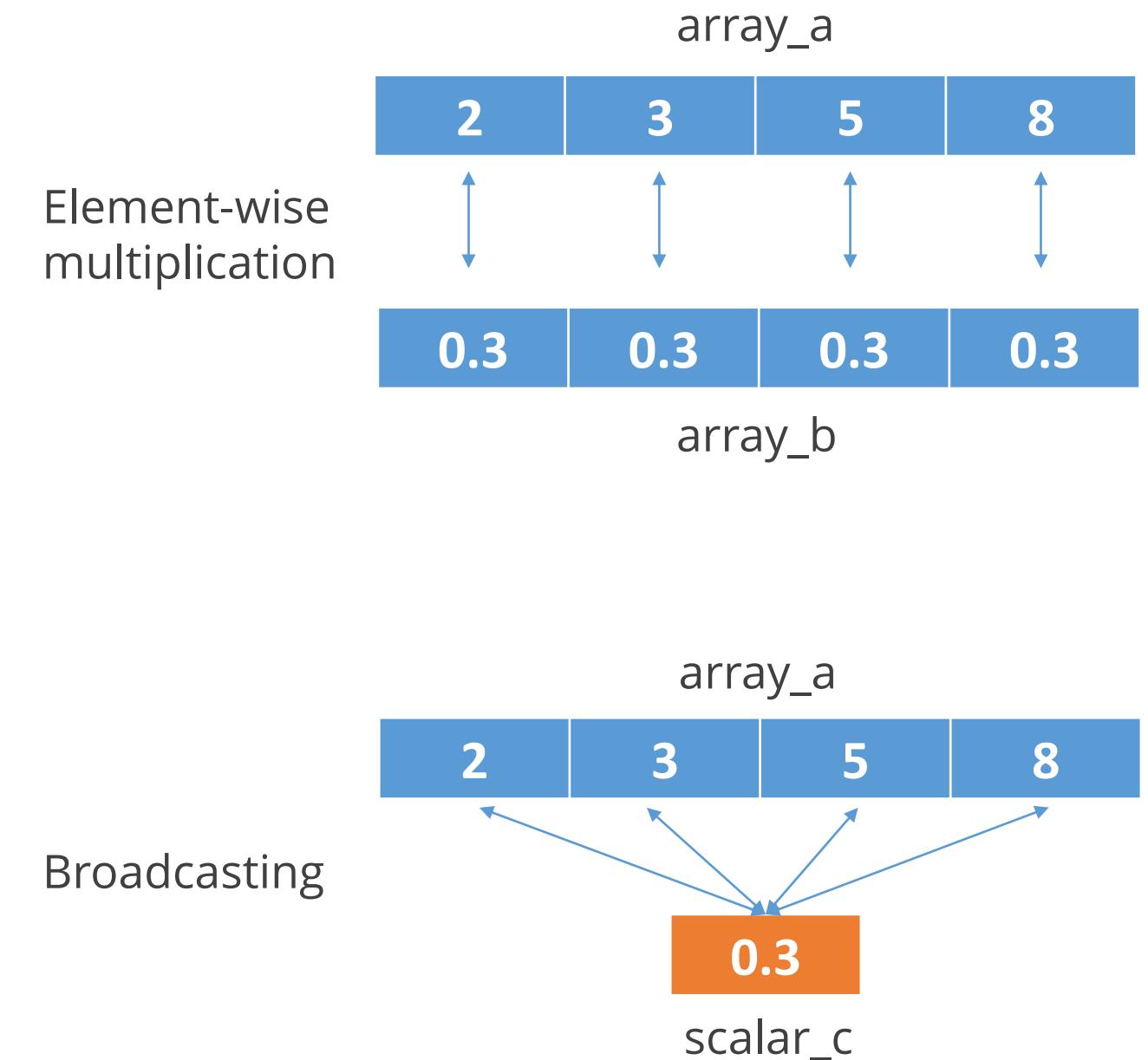
```
In [391]: np.hstack((new_cyclist_1,new_cyclist_2)) ← Stacks the arrays together
```

```
Out[391]: array([10, 15, 17, 26, 13, 19, 12, 11, 21, 24, 14, 23])
```

# Broadcasting

NumPy uses broadcasting to carry out arithmetic operations between arrays of different shapes. In this method, NumPy automatically broadcasts the smaller array over the larger array.

```
In [9]: import numpy as np  
  
In [10]: #Create two arrays of the same shape  
array_a = np.array([2, 3, 5, 8])  
array_b = np.array([.3, .3, .3, .3])  
  
In [11]: #Multiply arrays  
array_a * array_b  
  
Out[11]: array([ 0.6,  0.9,  1.5,  2.4])  
  
In [12]: #Create a variable with a scalar value  
scalar_c = .3  
  
In [13]: #Multiply 1D array with a scalar value  
array_a * scalar_c  
  
Out[13]: array([ 0.6,  0.9,  1.5,  2.4])
```



# Broadcasting—Constraints

Though broadcasting can help carry out mathematical operations between different-shaped arrays, they are subject to certain constraints as listed below:

```
In [9]: import numpy as np
```

```
In [10]: #Create two arrays of the same shape  
array_a = np.array([2, 3, 5, 8])  
array_b = np.array([.3, .3, .3, .3])
```

```
In [11]: #Multiply arrays  
array_a * array_b
```

```
Out[11]: array([ 0.6,  0.9,  1.5,  2.4])
```

```
In [14]: #Create array of a different shape  
array_d = np.array([4, 3])
```

```
In [15]: array_a * array_d
```

```
-----  
ValueError
```

```
<ipython-input-15-43adcf6f7a54> in <module>()  
----> 1 array_a * array_d
```

```
Traceback (most recent call last)
```

```
ValueError: operands could not be broadcast together with shapes (4,) (2,)
```

- When NumPy operates on two arrays, it compares their shapes element-wise. It finds these shapes compatible only if:
  - Their dimensions are the same or
  - One of them has a dimension of size 1.
- If these conditions are not met, a "ValueError" is thrown, indicating that the arrays have incompatible shapes.

# Broadcasting—Example

Let's look at an example to see how broadcasting works to calculate the number of working hours of a worker per day in a certain week.

```
In [246]: np_week_one = np.array([105, 135, 195, 120, 165]) ← Week one earnings  
          np_week_two = np.array([123, 156, 230, 200, 147]) ← Week two earnings
```

```
In [247]: total_earning = np_week_one+np_week_two
```

```
In [248]: total_earning
```

```
Out[248]: array([228, 291, 425, 320, 312]) ← Total earning for 2 weeks
```

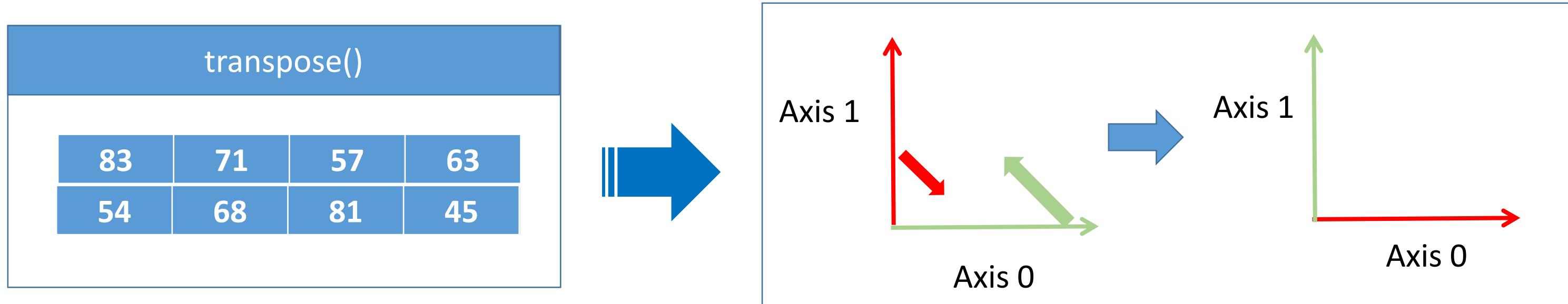
```
In [249]: np_week_one_hrs = np_week_one / 15 ← Calculate week one hours
```

```
In [250]: np_week_one_hrs
```

```
Out[250]: array([ 7,  9, 13,  8, 11]) ← Number of working hours  
                    per day in week one
```

# Linear Algebra—Transpose

NumPy can carry out linear algebraic functions as well. The “transpose()” function can help you interchange rows as columns, and vice-versa.



```
In [397]: test_scores = np.array([[83, 71, 57, 63], [54, 68, 81, 45]])
```

```
In [398]: test_scores.transpose()
```

```
Out[398]: array([[83, 54],  
                  [71, 68],  
                  [57, 81],  
                  [63, 45]])
```

# Linear Algebra—Inverse and Trace Functions

Using NumPy, you can also find the inverse of an array and add its diagonal data elements.

## np.linalg.inv()

```
In [411]: inverse_array =np.array([[10,20],[15,25]])
```

```
In [412]: np.linalg.inv(inverse_array)
```

```
Out[412]: array([[-0.5,  0.4],  
                  [ 0.3, -0.2]])
```

Inverse of the given array

\* Can be applied only on a square matrix

## np.trace()

```
In [420]: trace_array =np.array([[10,20],[22,31]])
```

```
In [421]: np.trace(trace_array)
```

```
Out[421]: 41
```

Sum of diagonal elements “10” and “31”



# Assignment

Problem

Instructions

Evaluate the dataset containing the GDPs of different countries to:

- Find and print the name of the country with the highest GDP,
- Find and print the name of the country with the lowest GDP,
- Print out text and input values iteratively,
- Print out the entire list of the countries with their GDPs, and
- Print the highest GDP value, lowest GDP value, mean GDP value, standardized GDP value, and the sum of all the GDPs.

Problem

Instructions

Instructions to perform the assignment:

- Download the GDP dataset from the “Resource” tab. You can copy the data provided to help you with your assignment.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



# Assignment

Problem

Instructions

Evaluate the dataset of the Summer Olympics, London 2012 to:

- Find and print the name of the country that won maximum gold medals,
- Find and print the countries who won more than 20 gold medals,
- Print the medal tally,
- Print each country name with the corresponding number of gold medals, and
- Print each country name with the total number of medals won.

Problem

Instructions

Instructions to perform the assignment:

- Download the “Olympic 2012 Medal Tally” dataset. Use the data provided to create relevant and required variables.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



**QUIZ****1**

**Which of the following arrays is valid?**

- a. [1, 0.3, 8, 6.4]
- b. ["Lucy", 16, "Susan", 23, "Carrie", 37]
- c. [True, False, "False", True]
- d. [3.14j, 7.3j, 5.1j, 2j]



**QUIZ****1****Which of the following arrays is valid?**

- a. [1, 0.3, 8, 6.4]
- b. ["Lucy", 16, "Susan", 23, "Carrie", 37]
- c. [True, False, "False", True]
- d. [3.14j, 7.3j, 5.1j, 2j]



The correct answer is **d**.

**Explanation:** A NumPy ndarray can hold only a single data type, which makes it homogenous. NumPy supports integers, floats, Booleans, and even complex numbers. Of all the options provided, only the array containing complex numbers is homogenous. All the other options contain more than one data type.

**QUIZ**  
**2**

**Which function is most useful to convert a multidimensional array into a one-dimensional array?**

- a. ravel()
- b. reshape()
- c. resize() and reshape()
- d. All of the above



**QUIZ**  
**2**

**Which function is most useful to convert a multidimensional array into a one-dimensional array?**

- a. ravel()
- b. reshape()
- c. resize() and reshape()
- d. All of the above



The correct answer is **a.**

**Explanation:** The function ravel() is used to convert a multidimensional array into a one-dimensional array. Though reshape() also functions in a similar way, it creates a new array instead of transforming the input array.

**QUIZ****3**

The np.trace() method gives the sum of \_\_\_\_.

- a. the entire array
- b. the diagonal elements from left to right
- c. the diagonal elements from right to left
- d. consecutive rows of an array



**QUIZ****3**

The np.trace() method gives the sum of \_\_\_\_.

- a. the entire array
- b. the diagonal elements from left to right
- c. the diagonal elements from right to left
- d. consecutive rows of an array



The correct answer is **b**.

**Explanation:** The trace() function is used to find the sum of the diagonal elements in an array. It is carried out in an incremental order of the indices. Therefore, it can only add diagonal values from left to right and not vice versa.

**QUIZ****4**

**The function np.transpose() when applied on a one-dimensional array gives \_\_\_\_.**

- a. a reverse array
- b. an unchanged original array
- c. an inverse array
- d. all elements with zeroes



## QUIZ

4

The function np.transpose() when applied on a one dimensional array gives \_\_\_\_.

- a. a reverse array
- b. an unchanged original array
- c. an inverse array
- d. all elements with zeroes



The correct answer is **b**.

**Explanation:** Transposing a one-dimensional array does not change it in any way. It returns an unchanged view of the original array.

**QUIZ**  
**5**

**Which statement will slice the highlighted data?**

11 | 14 | 21 | 32 | 53 | 64

- a. [3 : 5]
- b. [3 : 6]
- c. [2 : 5]
- d. [2 : 4]



**QUIZ**  
**5**

**Which statement will slice the highlighted data?**

11 | 14 | 21 | 32 | 53 | 64

- a. [3 : 5]
- b. [3 : 6]
- c. [2 : 5]
- d. [2 : 4]

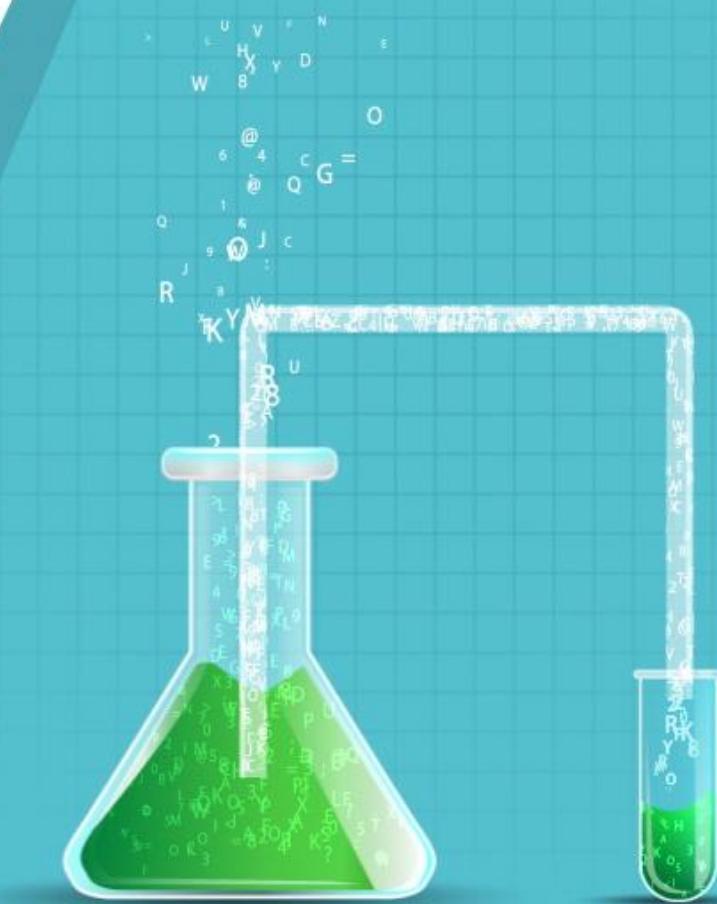


The correct answer is **c**.

**Explanation:** Let's assume that the index of the first element is  $m$  and the second element is  $n$ . Then, you need to use the statement “[ $n : m + 1$ ]” to slice the required dataset. In this case, the index of the element “21” is “2” and that of “53” is “4.” So, the correct statement to use would be [2 : 5].

# Key Takeaways

- NumPy is a very powerful Python library for mathematical and scientific computing.
- You can create and print NumPy arrays using different methods.
- Arrays can be one-dimensional, two-dimensional, three-dimensional, or multi-dimensional.
- NumPy uses basic operations, data access techniques, and copy and view techniques for data wrangling.
- NumPy can also manipulate data using various array shape manipulation techniques.
- NumPy can perform linear algebra functions to fix problematic datasets and execute mathematical operations.



**This concludes “Mathematical Computing with Python (NumPy).”**  
The next lesson is “Scientific Computing with Python (SciPy).”

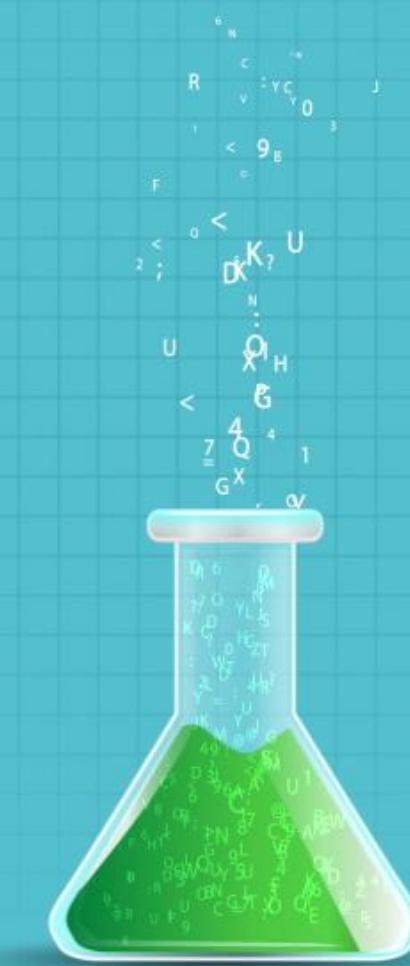


## Data Science with Python

### Lesson 06—Python: Scientific computing with Python (SciPy)

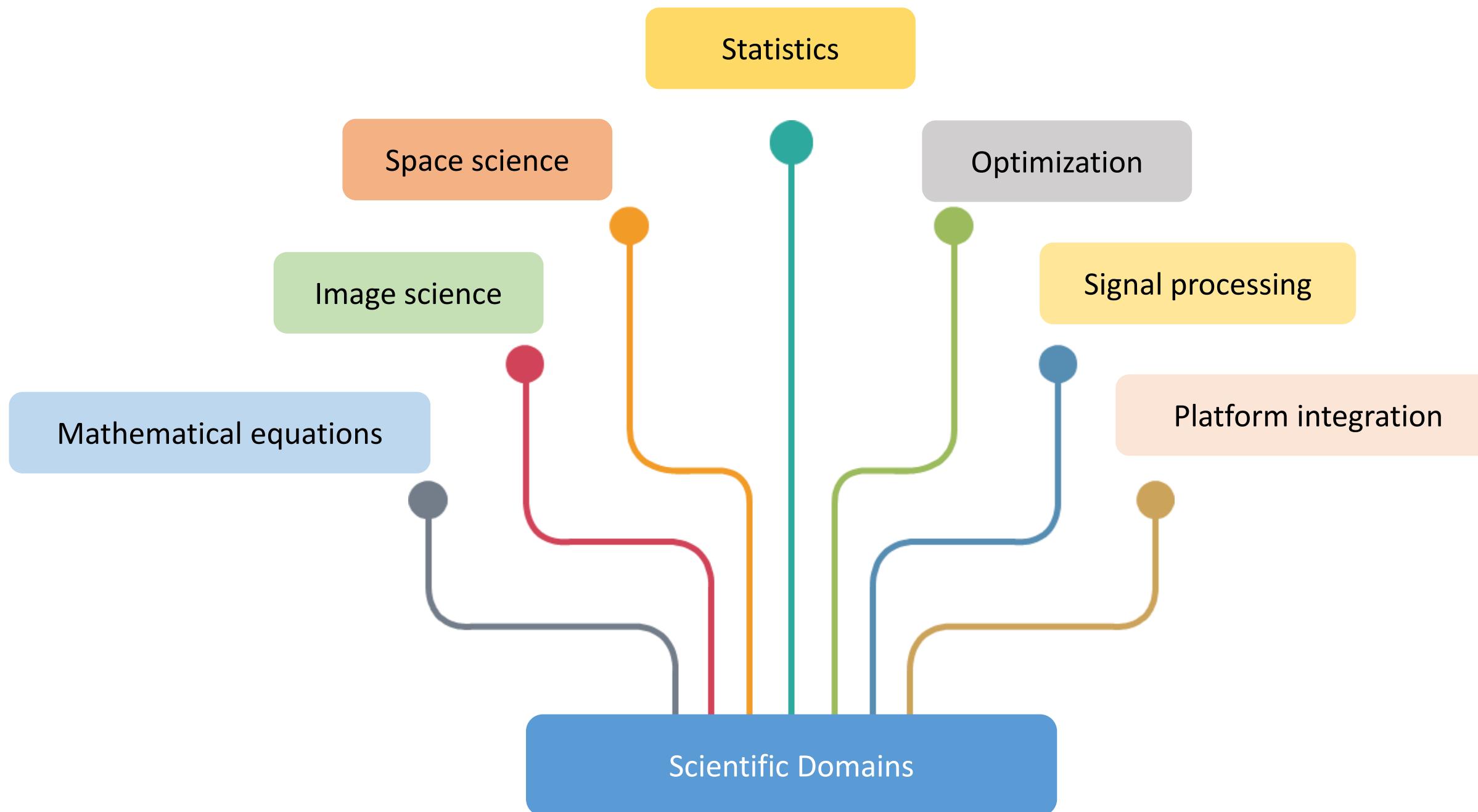
# What You'll Learn

- Why SciPy is needed
- The characteristics of SciPy
- The sub-packages of SciPy
- SciPy Sub-packages such as Optimization, Integration, Linear Algebra, Statistics, Weave, and IO



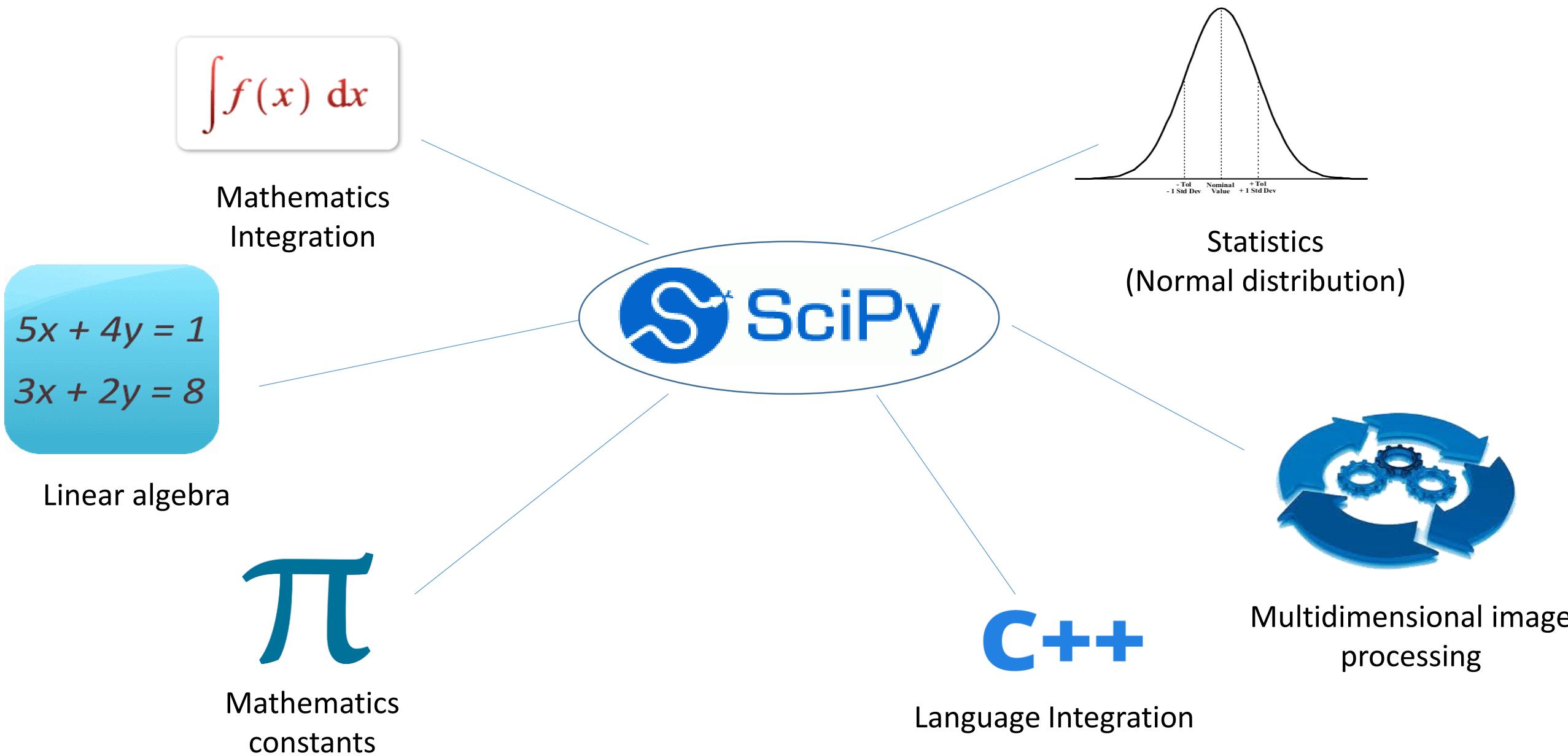
# The Real World: Multiple Scientific Domains

How to handle multiple scientific domains? The solution is SciPy.



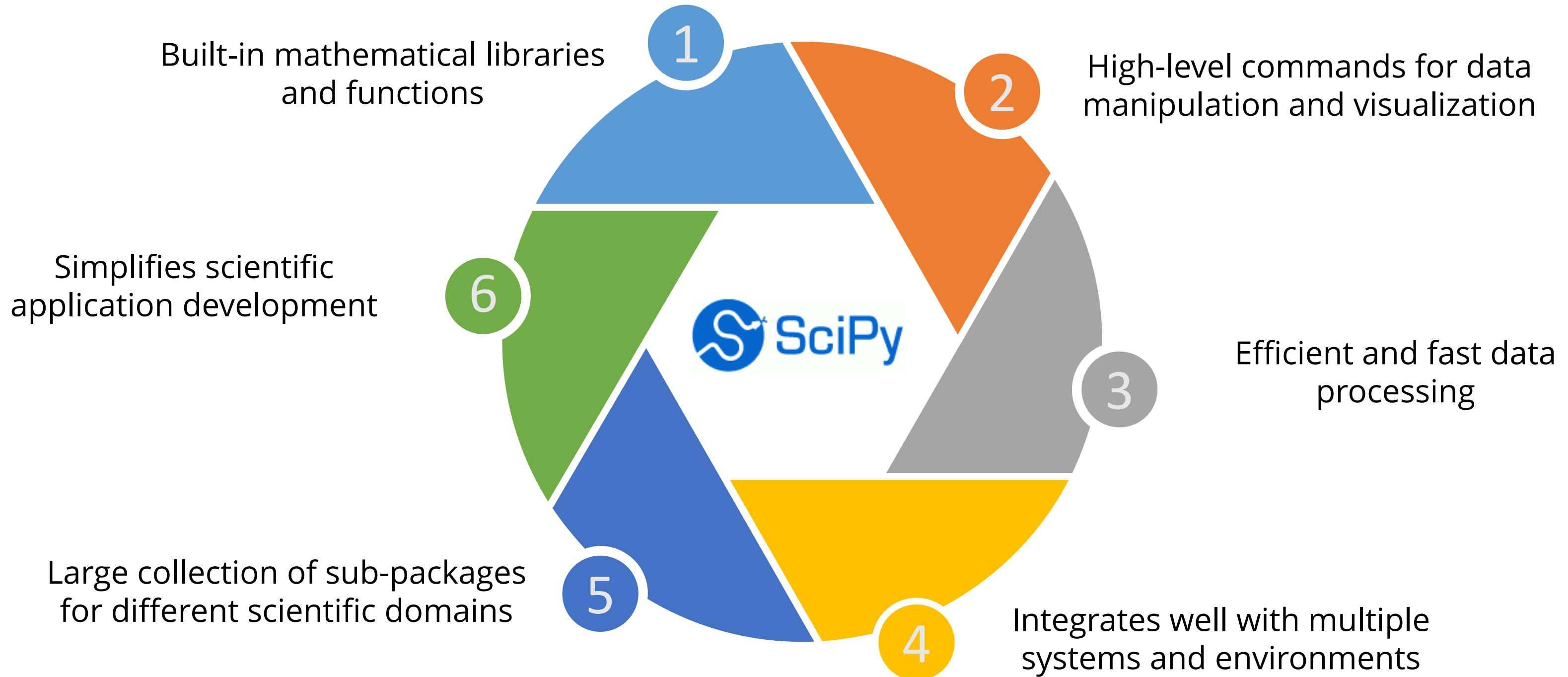
# SciPy: The Solution

SciPy has built-in packages that help in handling the scientific domains.



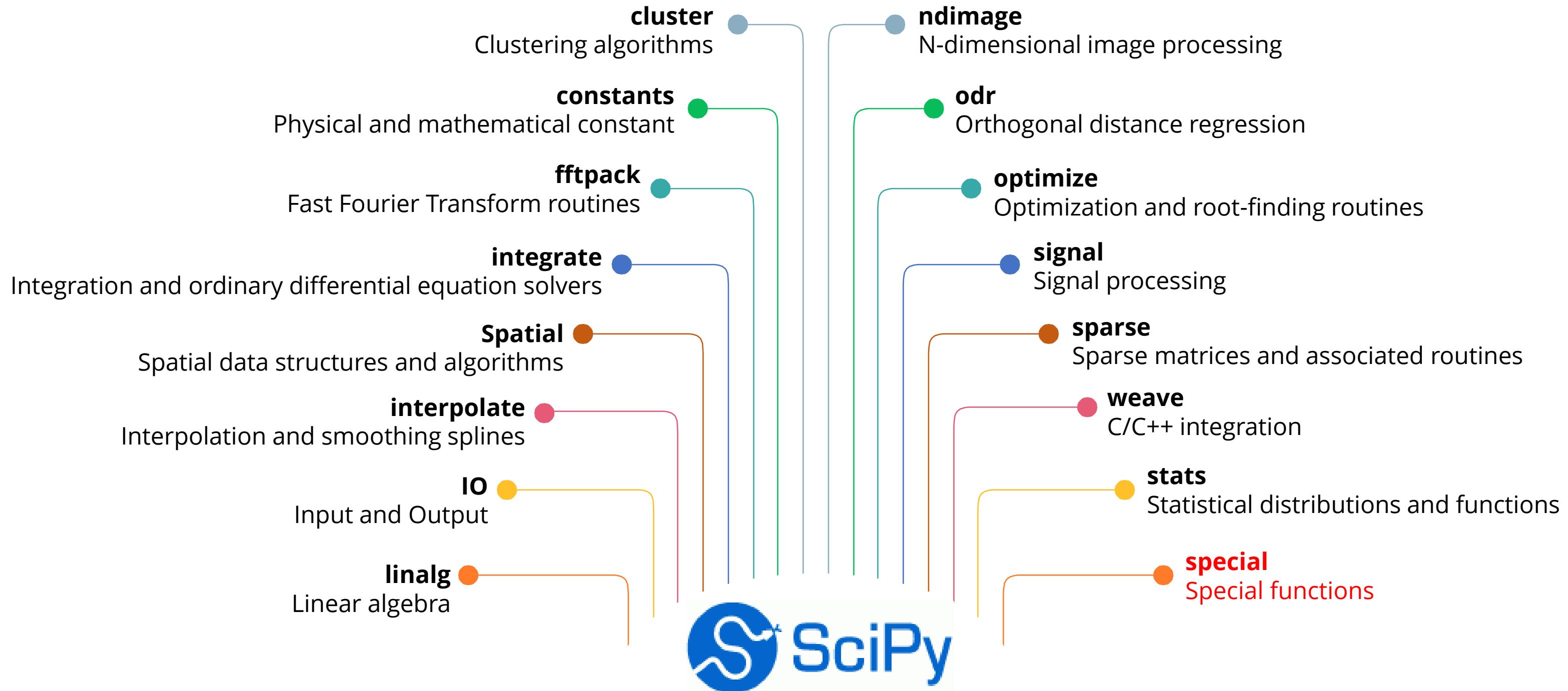
# SciPy and its Characteristics

Characteristics of SciPy are as follows:



# SciPy Sub-package

SciPy has multiple sub-packages which handle different scientific domains.

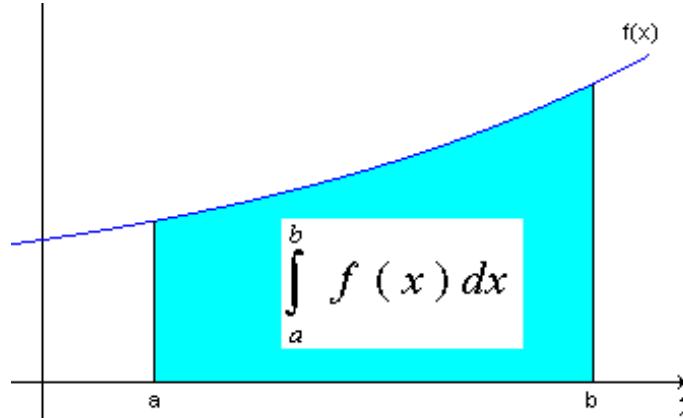


# SciPy Sub-package: Integration

SciPy provides integration techniques that solve mathematical sequences and series, or perform function approximation.

## General integration (quad)

- `integrate.quad(f, a, b)`



## General multiple integration (dblquad, tplquad, nquad)

- `integrate.dblquad()`
- `integrate.tplquad()`
- `integrate.nquad()`

The limits of all inner integrals need to be defined as functions.

# SciPy Sub-package: Integration

This example shows how to perform quad integration.

In [13]: `from scipy.integrate import quad`

In [14]: `def integrateFunction(x):  
 return x`

In [15]: `quad(integrateFunction,0,1)`

Out[15]: `(0.5, 5.551115123125783e-15)`

In [16]: `def integrateFn(x,a,b):  
 return x*a+b`

In [17]: `a=3  
b=2`

In [18]: `quad(integrateFn,0,1,args=(a,b))`

Out[18]: `(3.5, 3.885780586188048e-14)`

Import quad from integrate sub-package

Define function for integration of x

Perform quad integration for function of x for limit 0 to 1

Define function for  $ax + b$

Declare value of a and b

Perform quad integration and pass functions and arguments

## SciPy Sub-package: Integration

This example shows you how to perform multiple integration.

In [20]: `import scipy.integrate as integrate`

Import integrate package  
sub-package

In [21]: `def f(x, y):  
 return x + y  
integrate.dblquad(f, 0, 1,lambda x: 0, lambda x: 2)`

Define function for  $x + y$

Out[21]: `(3.0, 3.3306690738754696e-14)`

Perform multiple  
integration using the  
lambda built-in function

# SciPy Sub-package: Optimization

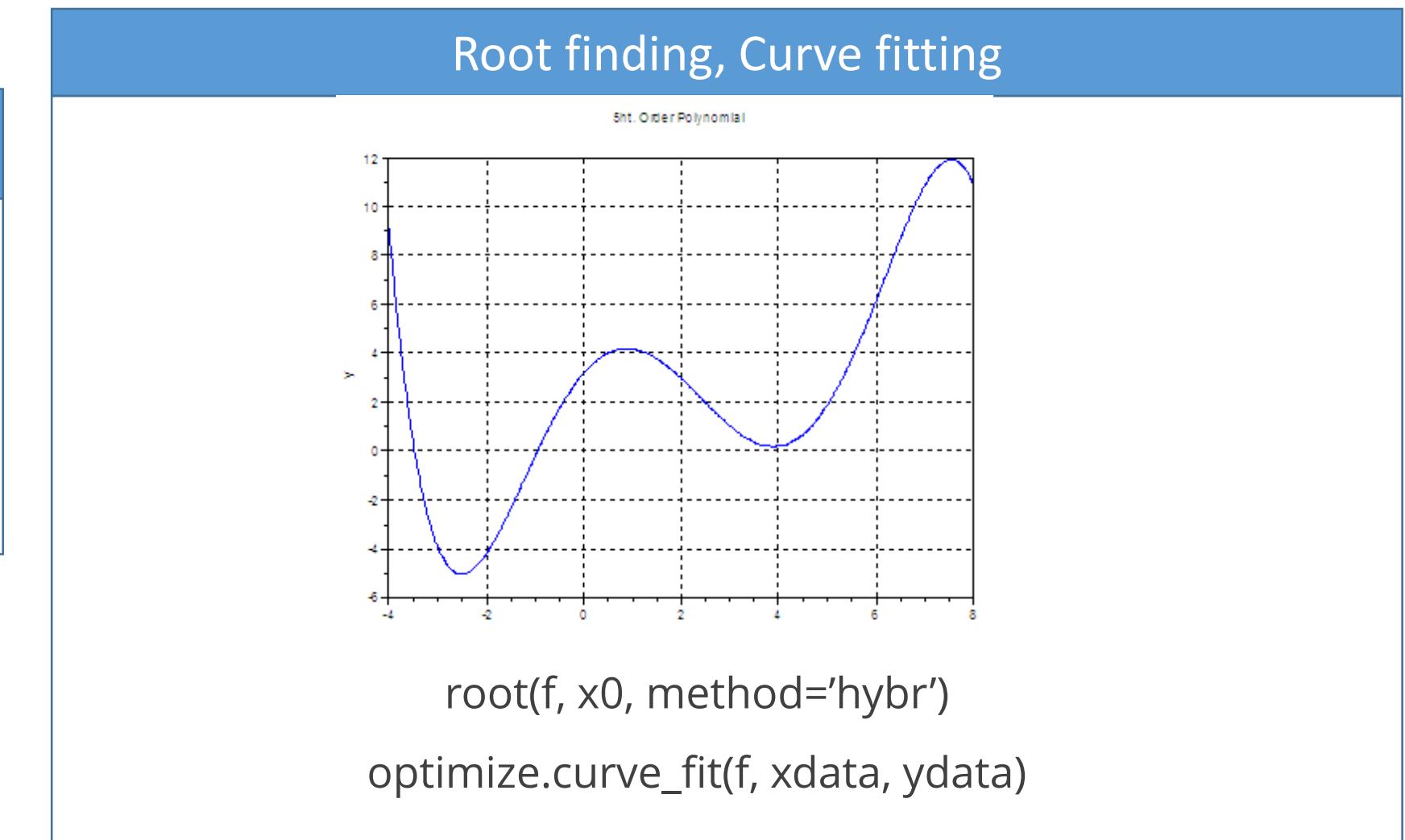
Optimization is a process to improve performance of a system mathematically by fine-tuning the process parameters.

SciPy provides several optimization algorithms such as bfgs, Nelder-Mead simplex, Newton Conjugate Gradient, COBYLA, or SLSQP.

## Minimization functions

```
optimize.minimize(f, x0, method='BFGS')
```

lower limit in a given range



# SciPy Sub Package: Optimization

```
In [32]: import numpy as np  
from scipy import optimize
```

Import numpy and  
optimize from scipy

```
In [33]: def f(x):  
    return x**2 + 5*np.sin(x)
```

Define function for  
 $X^2 + 5 \sin x$

```
In [34]: minimaValue = optimize.minimize(f,x0=2,method='bfgs',options={'disp':True})
```

Optimization terminated successfully.  
Current function value: -3.246394  
Iterations: 4  
Function evaluations: 24  
Gradient evaluations: 8

Perform optimize  
minimize function  
using bfgs method  
and options

```
In [35]: minimaValueWithoutOpt = optimize.minimize(f,x0=2,method='bfgs')
```

```
In [36]: minimaValueWithoutOpt
```

```
Out[36]: {'fun': -3.2463942726915382,  
          'hess_inv': array([[ 0.15430551]]),  
          'jac': array([-8.94069672e-08]),  
          'message': 'Optimization terminated successfully.',  
          'nfev': 24,  
          'nit': 4,  
          'njev': 8,  
          'status': 0,  
          'success': True,  
          'x': array([-1.11051051])}
```

Perform optimize minimize  
function using bfgs method  
and without options

## SciPy Sub-package: Optimization

```
In [118]: import numpy as np
from scipy.optimize import root
def rootfunc(x):
    return x + 3.5 * np.cos(x)
```

Define function for X + 3.5 Cos x

```
In [119]: rootValue = root(rootfunc, 0.3)
```

Pass x value in argument for root

```
In [120]: rootValue
```

```
Out[120]: fjac: array([[-1.]])
        fun: array([ 2.22044605e-16])
      message: 'The solution converged.'
       nfev: 14
         qtf: array([-8.32889313e-13])
           r: array([-4.28198145])
       status: 1
     success: True
          x: array([-1.21597614])
```

Function value and array values



# Knowledge Check

KNOWLEDGE  
CHECK

What are the specification limits provided for curve fitting function (**optimize.curve.fit**), during the optimization process?

- a. Upper limit value
- b. Lower limit value
- c. Upper and lower limit values
- d. Only the optimization method



KNOWLEDGE  
CHECK

What are the specification limits provided for curve fitting function (**optimize.curve.fit**), during the optimization process?

- a. Upper limit value
- b. Lower limit value
- c. • Upper and lower limit values
- d. Only the optimization method



The correct answer is

• c

**Explanation:** Both the upper and lower limit values should be specified for **optimize.curve.fit** function.

# SciPy Sub-package: Linear Algebra

SciPy provides rapid linear algebra capabilities and contains advanced algebraic functions.

*Click each tab to know more.*

Inverse of matrix

Determinant

Linear systems

Single value  
decomposition (svd)

This function is used to compute the inverse of the given matrix. Let's take a look at the inverse matrix operation.

```
In [65]: import numpy as np
from scipy import linalg
matrix = np.array([[10,6],[2,7]])
matrix
```

Import linalg and Define a numpy matrix or array

```
Out[65]: array([[10, 6],
                 [ 2, 7]])
```

```
In [66]: type(matrix)
```

```
Out[66]: numpy.ndarray
```

View the type

```
In [67]: linalg.inv(matrix)
```

```
Out[67]: array([[ 0.12068966, -0.10344828],
                 [-0.03448276,  0.17241379]])
```

Use inv function to inverse the matrix

# SciPy Sub-package: Linear Algebra

SciPy provides very rapid linear algebra capabilities and contains advanced algebraic functions.

*Click each tab to know more.*

Inverse of matrix

Determinant

Linear systems

Single value  
decomposition (svd)

With this function you can compute the value of the determinant for the given matrix.

```
In [68]: import numpy as np
from scipy import linalg
matrix = np.array([[4,9],[3,5]])
matrix
```

Import linalg and  
Define an numpy matrix or  
array

```
Out[68]: array([[4, 9],
                 [3, 5]])
```

```
In [69]: linalg.det(matrix)
```

Use det function to find the  
determinant value of the  
matrix

```
Out[69]: -7.00000000000001
```

# SciPy Sub-package: Linear Algebra

SciPy provides very rapid linear algebra capabilities and contains advanced algebraic functions.

*Click each tab to know more.*

Inverse of matrix

Determinant

Linear systems

Single value  
decomposition (svd)

## Linear equations

$$\begin{aligned} 2x + 3y + z &= 21 \\ -x + 5y + 4z &= 9 \\ 3x + 2y + 9z &= 6 \end{aligned}$$

```
In [88]: import numpy as np
          from scipy import linalg ← Import linalg

In [89]: numArray = np.array([[2,3,1],[-1,5,4],[3,2,9]])

In [90]: numArrValue = np.array([21,9,6])

In [91]: linalg.solve(numArray,numArrValue)
Out[91]: array([ 4.95,  4.35, -1.95])
```

Use solve method

# SciPy Sub-package: Linear Algebra

SciPy provides very rapid linear algebra capabilities and contains advanced algebraic functions.

*Click each tab to know more.*

Inverse of matrix

Determinant

Linear systems

Single value  
decomposition (svd)

```
In [103]: import numpy as np
from scipy import linalg
```

In [104]: numSvdArr = np.array([[3,5,1],[9,5,7]])

In [105]: numSvdArr.shape

Out[105]: (2L, 3L)

In [106]: linalg.svd(numSvdArr)

Out[106]: (array([[-0.37879831, -0.92547925],
 [-0.92547925, 0.37879831]]),
 array([[ 13.38464336, 3.29413449]]),
 array([[[-0.7072066 , -0.4872291 , -0.51231496],
 [ 0.19208294, -0.82977932, 0.52399467],
 [-0.68041382, 0.27216553, 0.68041382]]))

Import linalg

Define matrix

Find shape of ndarray which is 2X3 matrix

Use svd function

U (Unitary matrix)

Sigma or square root of eigenvalues

VH is values collected into unitary matrix

## Demo—Calculate Eigenvalues

Demonstrate how to calculate eigenvalues.





# Knowledge Check

KNOWLEDGE  
CHECK

Which of the following function is used for inverting the matrix?

- a. SciPy.special
- b. SciPy.linalg
- c. SciPy.signal
- d. SciPy.stats



KNOWLEDGE  
CHECK

Which of the following function is used for inverting the matrix?

- a. SciPy.special
- b. SciPy.linalg
- c. • SciPy.signal
- d. SciPy.stats

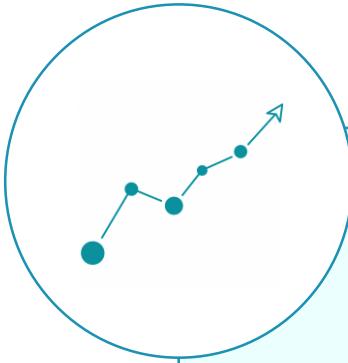


The correct answer is     • **b**

**Explanation:** SciPy.linalg is used to inverse the matrix.

## SciPy Sub-package: Statistics

SciPy provides a very rich set of statistical functions which are as follows:

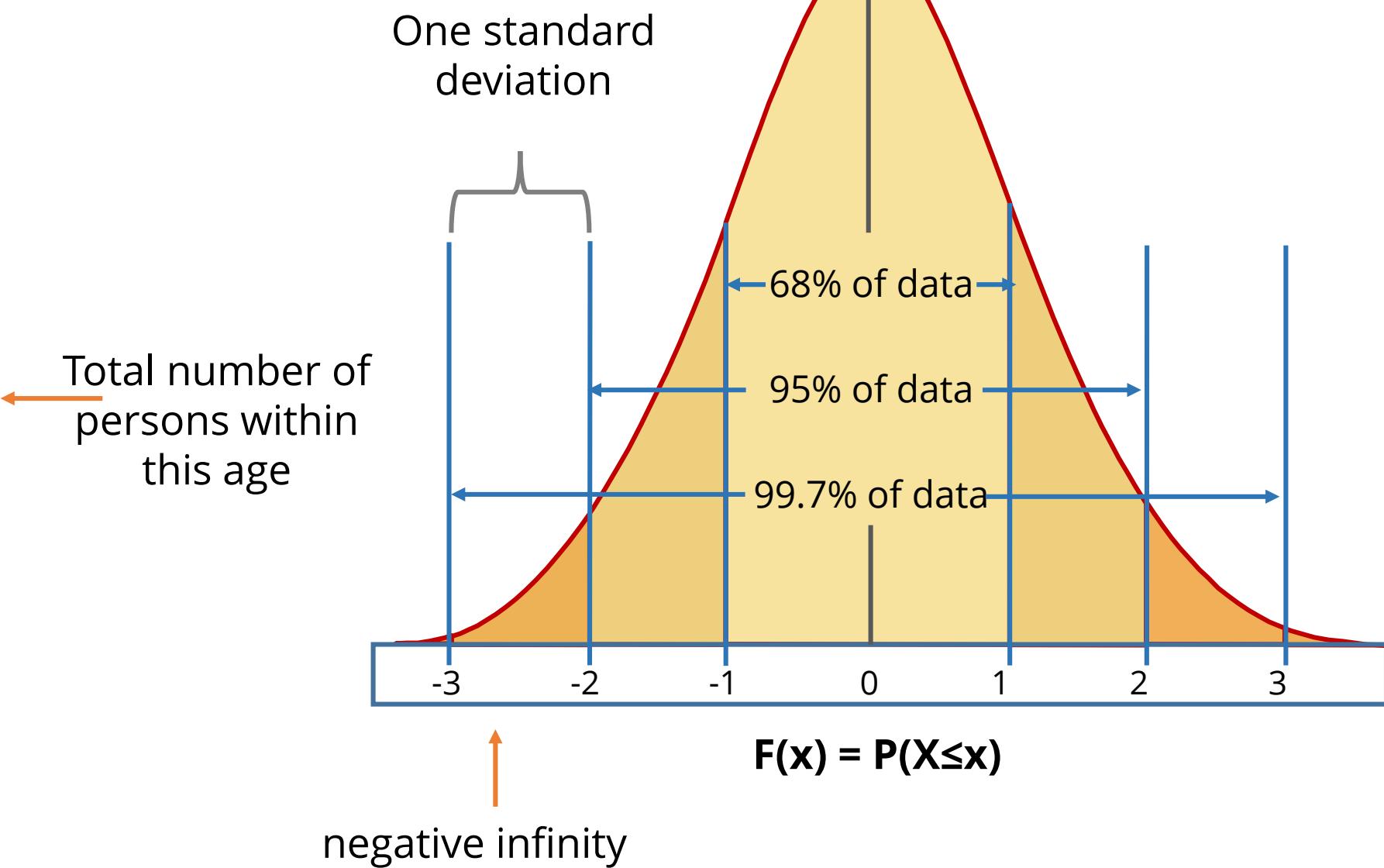


- This package contains distributions for which random variables are generated.
- These packages enable the addition of new routines and distributions. It also offers convenience methods such as pdf(), cdf()
- Following are the statistical functions for a set of data:
  - linear regression: linregress()
  - describing data: describe(), normaltest()

## SciPy Sub-package: Statistics

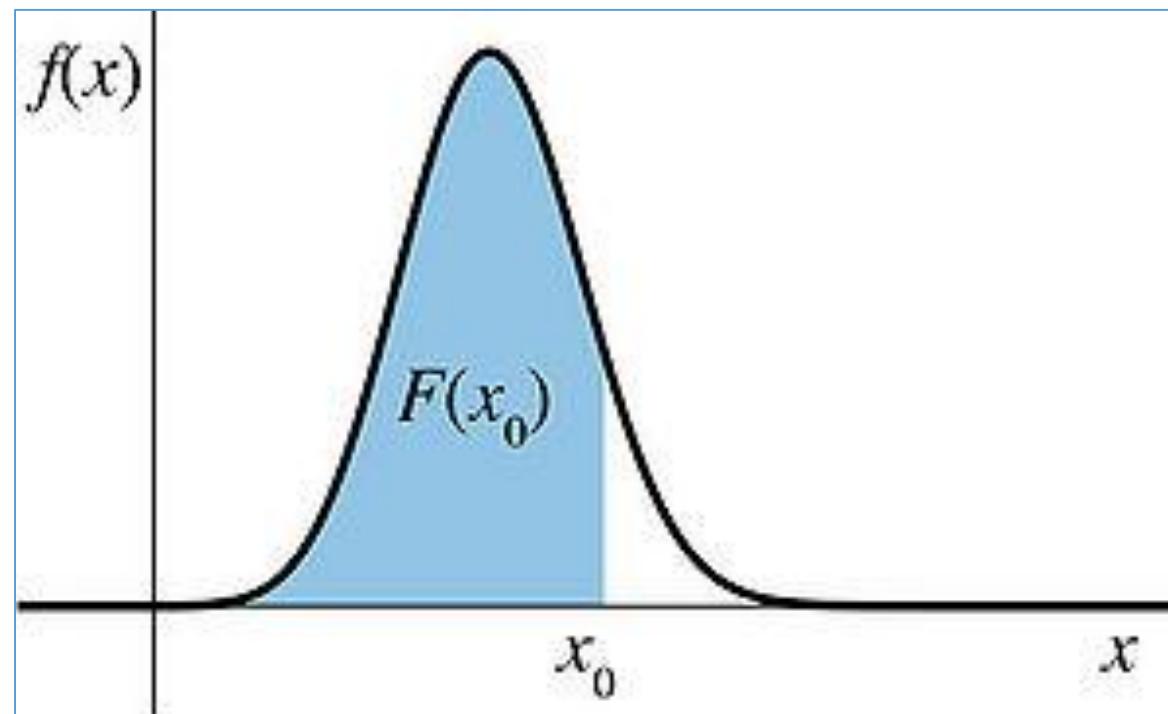
Cumulative Distribution Function provides the cumulative probability associated with a function.

Age Range	Frequency	Cumulative Frequency
0-10	19	19
10-20	55	74
21-30	23	97
31-40	36	133
41-50	10	143
51-60	17	160



## SciPy Sub-package: Statistics

Probability Density Function, or **PDF**, of a continuous random variable is the derivative of its Cumulative Distribution Function, or CDF.



$$f(x) = \frac{dF(x)}{dx}$$

Derivative of CDF

## SciPy Sub-package: Statistics

### Functions of Random Variables – Continuous (Normal Distribution):

```
In [108]: from scipy.stats import norm
```

← Import norm for normal distribution

```
In [110]: norm.rvs(loc=0,scale=1,size=10)
```

← rvs for Random variables

```
Out[110]: array([-0.16337774,  0.39039561,  0.85642826,  0.30134358, -1.86009474,
                   -0.29621603,  0.03863757,  0.23727056, -1.42395316, -0.5730162 ])
```

```
In [112]: norm.cdf(5,loc=1,scale=2)
```

← cdf for Cumulative Distribution Function

```
Out[112]: 0.97724986805182079
```

```
In [113]: norm.pdf(9,loc=0,scale=1)
```

← pdf for Probability Density  
Function for random  
distribution

```
Out[113]: 1.0279773571668917e-18
```



**loc** and **scale** are **used** to adjust the location and scale of the data distribution.

## SciPy Sub-package: Weave

The weave package provides ways to modify and extend any supported extension libraries.



- Includes C/C++ code within Python code
- Speed ups of 1.5x to 30x compared to algorithms written in pure Python

Two main functions of weave::

- `inline()` compiles and executes C/C++ code on the fly
- `blitz()` compiles NumPy Python expressions for fast execution

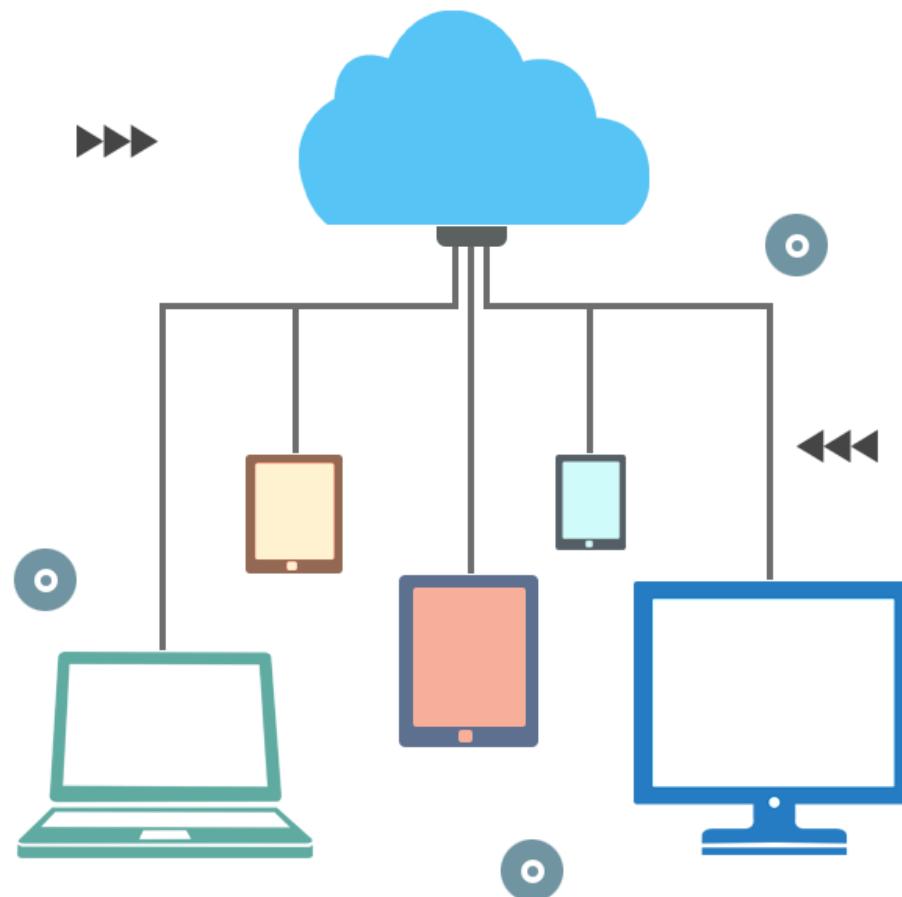
## SciPy Sub-package: IO

The IO package provides a set of functions to deal with several kinds of file formats.

It offers a set of functions to deal with file formats that includes:

- MatLab file
- IDL files
- Matrix Market files
- Wav sound files
- Arff files, and
- Netcdf files

Package provides additional files and it's corresponding methods.





Problem

Instructions

Use SciPy to solve a linear algebra problem.

There is a test with 30 questions worth 150 marks. The test has two types of questions:

1. True or false – carries 4 marks each
2. Multiple choice – carries 9 marks each

Find the number of true or false and multiple choice questions.

Problem

Instructions

## Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



Problem

Instructions

Use SciPy to declare 20 random values for random values and perform the following:

1. CDF – Cumulative Distribution Function for 10 random variables.
2. PDF – Probability Density Function for 14 random variables.

Problem

Instructions

## Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



**QUIZ  
1**

Which of the following is performed using SciPy?

- a. Website
- b. Plot data
- c. Scientific calculations
- d. System administration



QUIZ  
1

Which of the following is performed using SciPy?

- a. Website
- b. Plot data
- c. Scientific calculations
- d. System administration



The correct answer is **c.**

**Explanation:** SciPy has been specially made to perform scientific calculations. Generally, Python is the programming language that has libraries to perform all listed activities.

## QUIZ 2

Which of the following functions is used to calculate minima?

- a. optimize.minimize()
- b. integrate.quad()
- c. stats.linregress()
- d. linalg.solve()



QUIZ  
2

Which of the following functions is used to calculate minima?

- a. `optimize.minimize()`
- b. `integrate.quad()`
- c. `stats.linregress()`
- d. `linalg.solve()`



The correct answer is **a.**

**Explanation:** The function `optimize.minimize()` is used to calculate minima. `integrate.quad()` is used for integral calculation, `stats.linregress()` is used for linear regression, and `linalg.solve()` is used to solve a linear system.

QUIZ  
3

Which of the following syntaxes is used to generate 100 random variables from a t-distribution with df = 10?

- a. stats.t.pmf(df=10, size=100)
- b. stats.t.pdf(df=10, size=100)
- c. stats.t.rvs(df=10, size=100)
- d. stats.t.rand(df=10, size=100)



QUIZ  
3

Which of the following syntaxes is used to generate 100 random variables from a t-distribution with df = 10?

- a. stats.t.pmf(df=10, size=100)
- b. stats.t.pdf(df=10, size=100)
- c. stats.t.rvs(df=10, size=100)
- d. stats.t.rand(df=10, size=100)



The correct answer is **c**.

**Explanation:** The stats.t.rvs() function is used to generate random variables. stats.t.pmf() function is used to generate the probability of mass function, and stats.t.pdf() is used to generate probability density function. Note that stats.t.rand () does not exist.

# QUIZ

## 4

Which of the following functions is used to run C or C++ codes in SciPy?

- a. io.loadmat()
- b. weave.inline()
- c. weave.blitz()
- d. io.whosmat()



QUIZ  
4

Which of the following functions is used to run C or C++ codes in SciPy?

- a. io.loadmat()
- b. weave.inline()
- c. weave.blitz()
- d. io.whosmat()

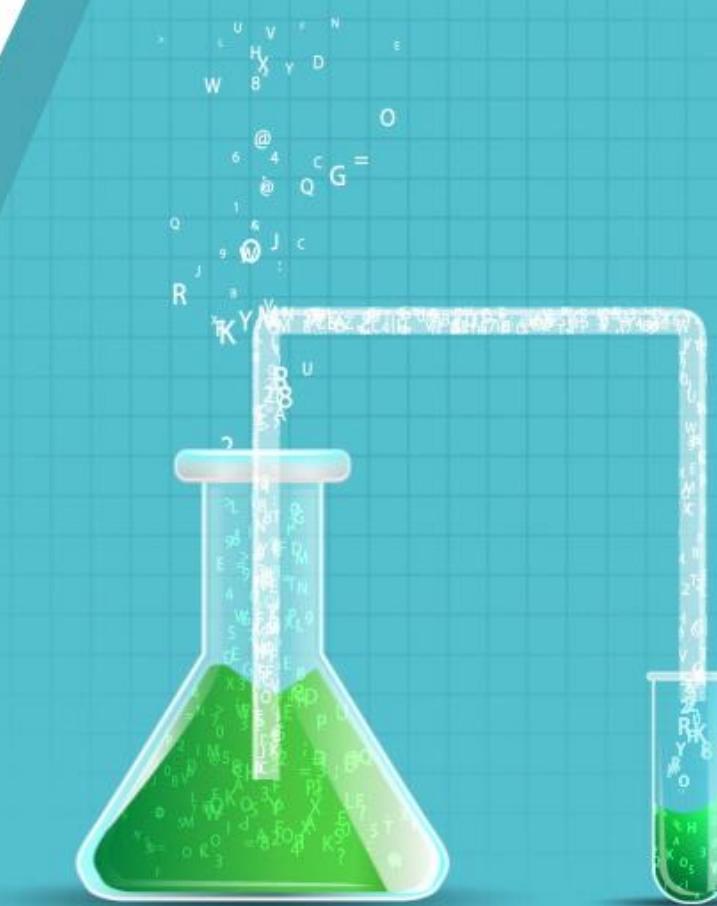


The correct answer is **b.**

Explanation: inline() function accepts C codes as string and compiles them for later use. loadmat() loads variables from .mat file. whosmat() checks the variables inside a .mat file. blitz() and then compiles NumPy expressions for faster running, but it can't accept C codes.

## Key Takeaways

- SciPy has multiple sub-packages, which proves useful for different scientific computing domains.
- Integration can be used to solve mathematical sequences and series or perform function approximation.
- Optimization is the process to improve performance of a system mathematically by fine-tuning the process parameters.
- The SciPy linear algebraic functions include computing the inverse of a matrix, calculating the determinant, solving linear systems, and computing single value decomposition.
- Statistical functions provide many useful sub-packages that enable the building of a hypothesis, determining the probability, and predicting the outcome.
- The IO package offers a set of functions to deal with several types of file formats.



**This concludes “Scientific computing with Python”**

The next lesson is “Data Manipulation with Pandas”

# Data Science with Python

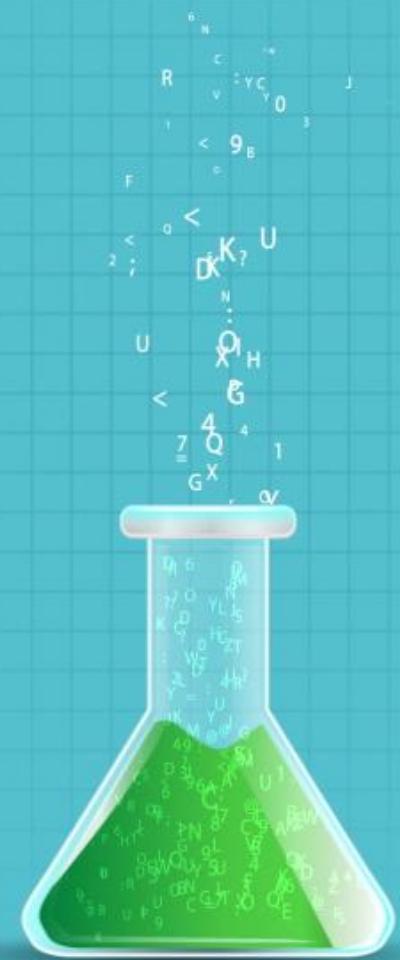
## Lesson 7— Data Manipulation with Pandas



# What You Will Learn

---

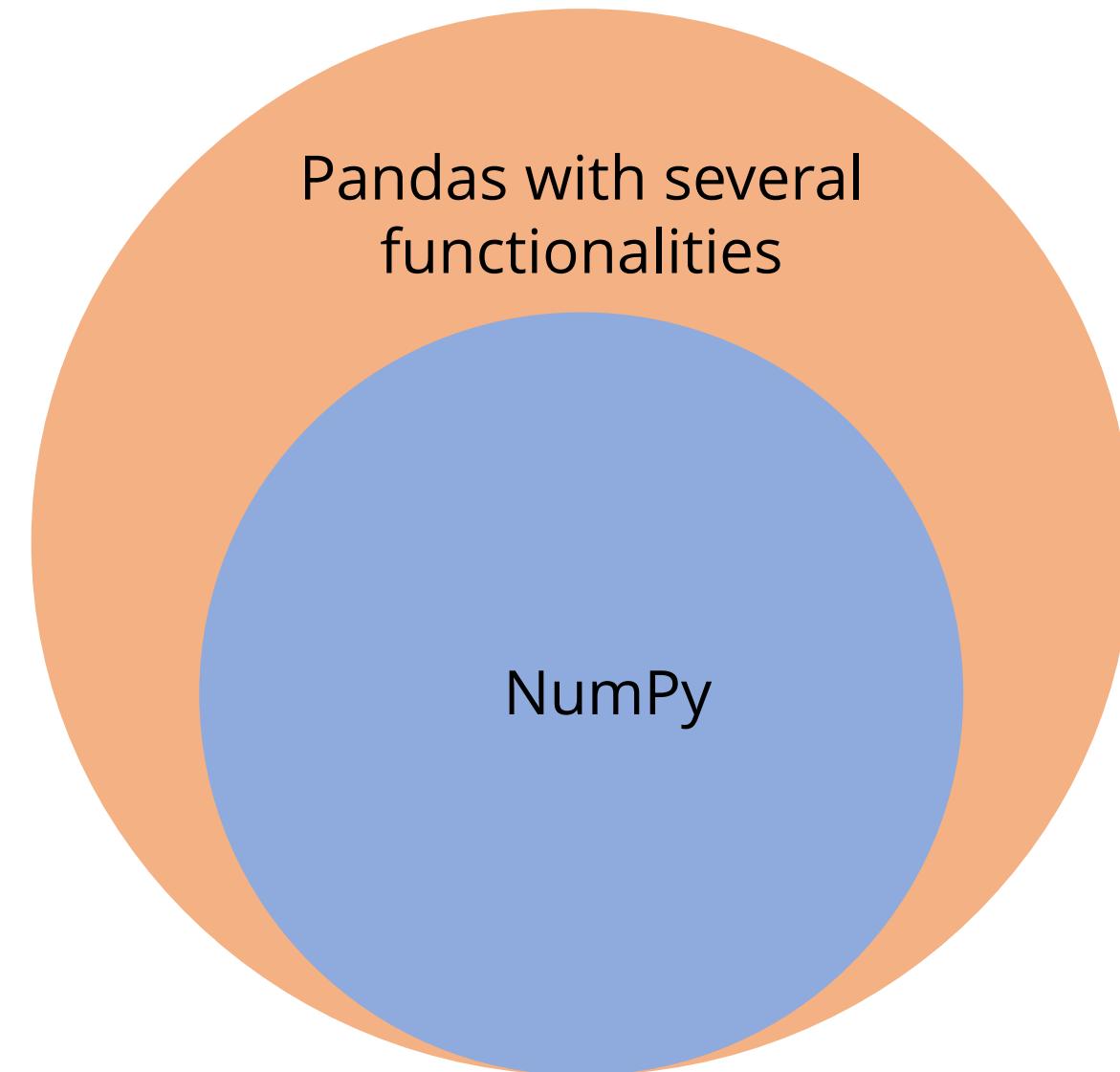
- Pandas and its features
- Different data structures of Pandas
- Creating Series and DataFrame with data inputs
- Viewing, selecting, and accessing elements in a data structure
- Handling vectorized operations
- Learning how to handle missing values
- Analyzing data with different data operation methods



# Why Pandas

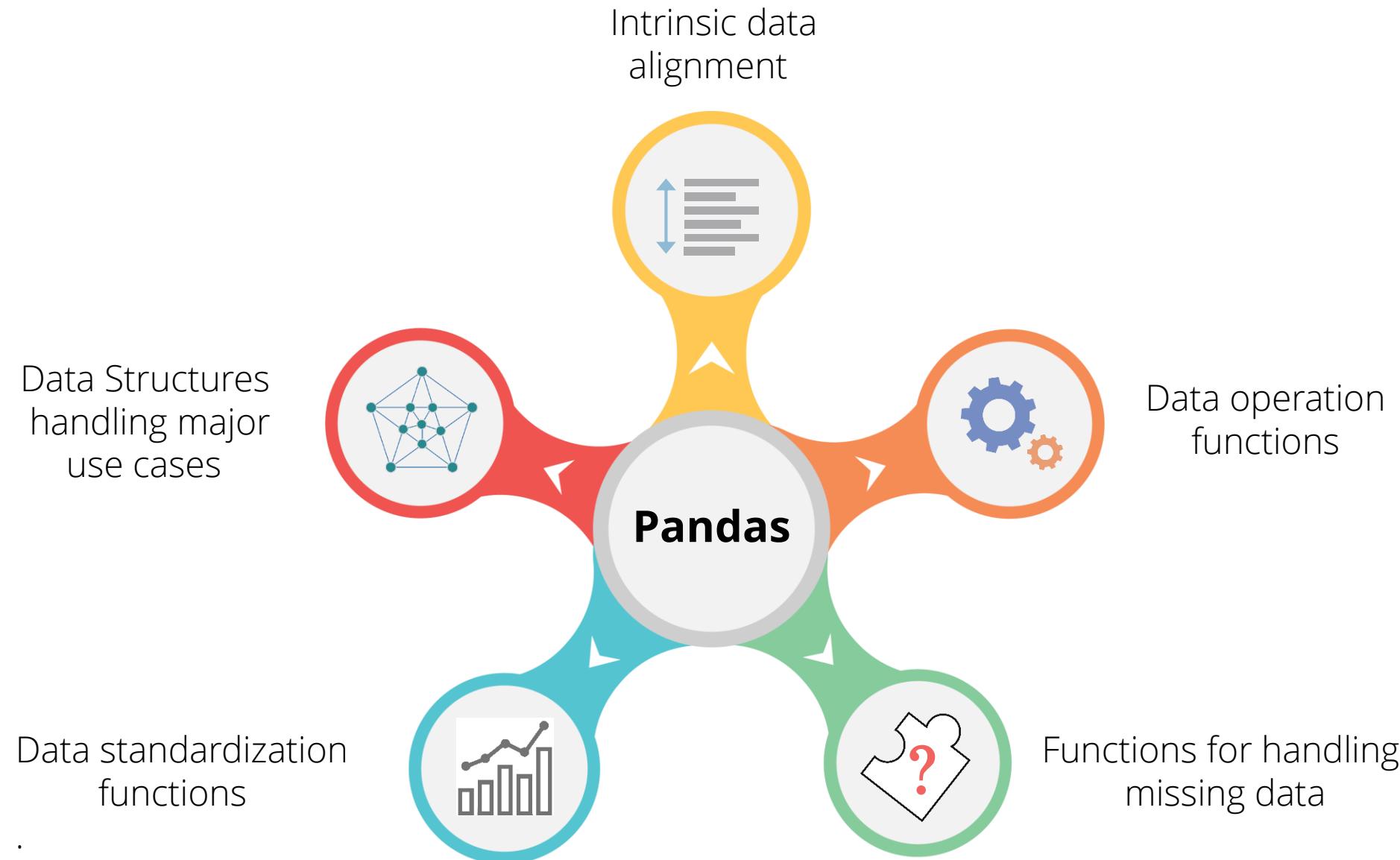
---

NumPy is great for mathematical computing. Then why do we need Pandas?



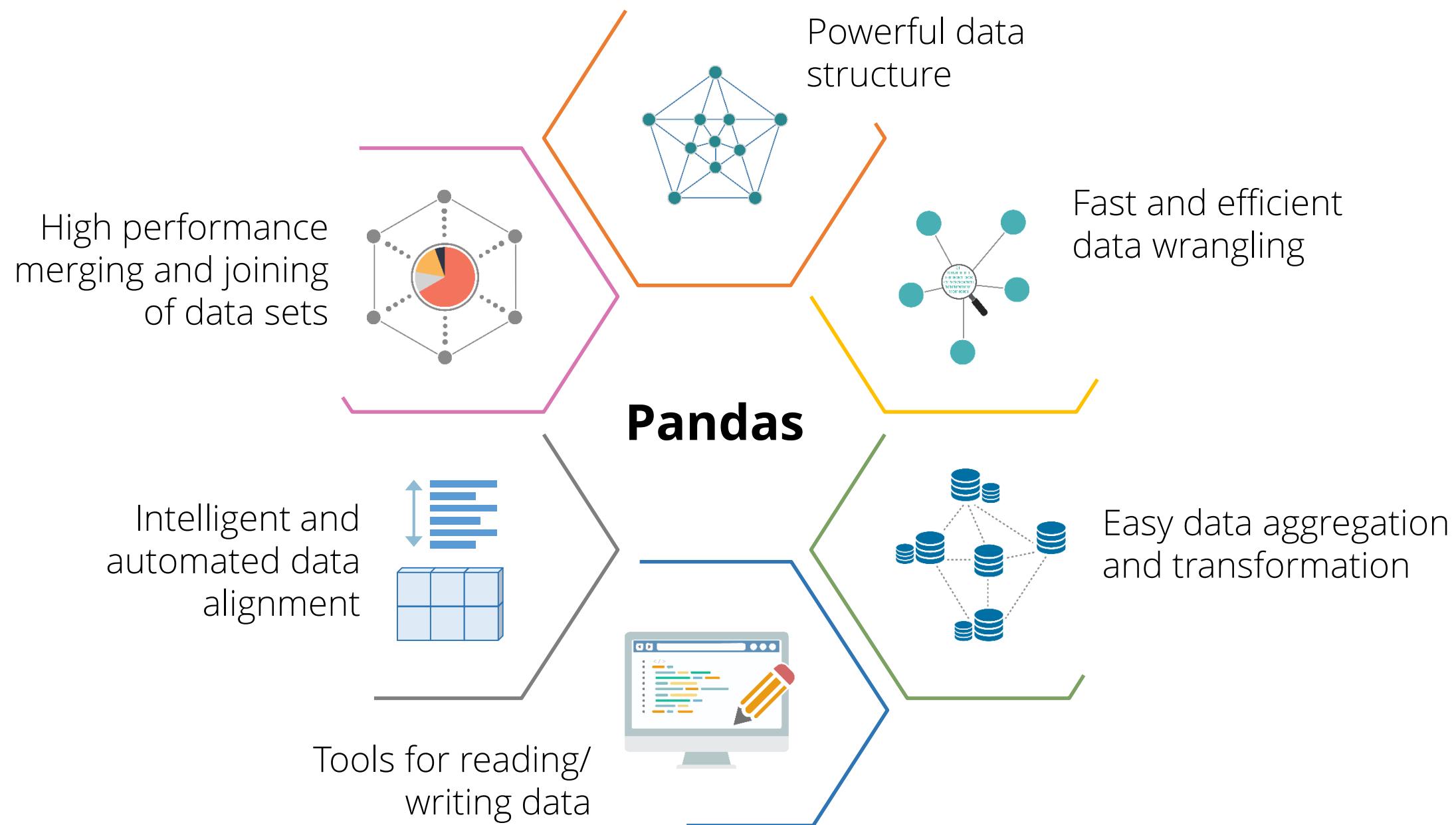
# Why Pandas

NumPy is great for mathematical computing. Then why do we need Pandas?



# Pandas Features

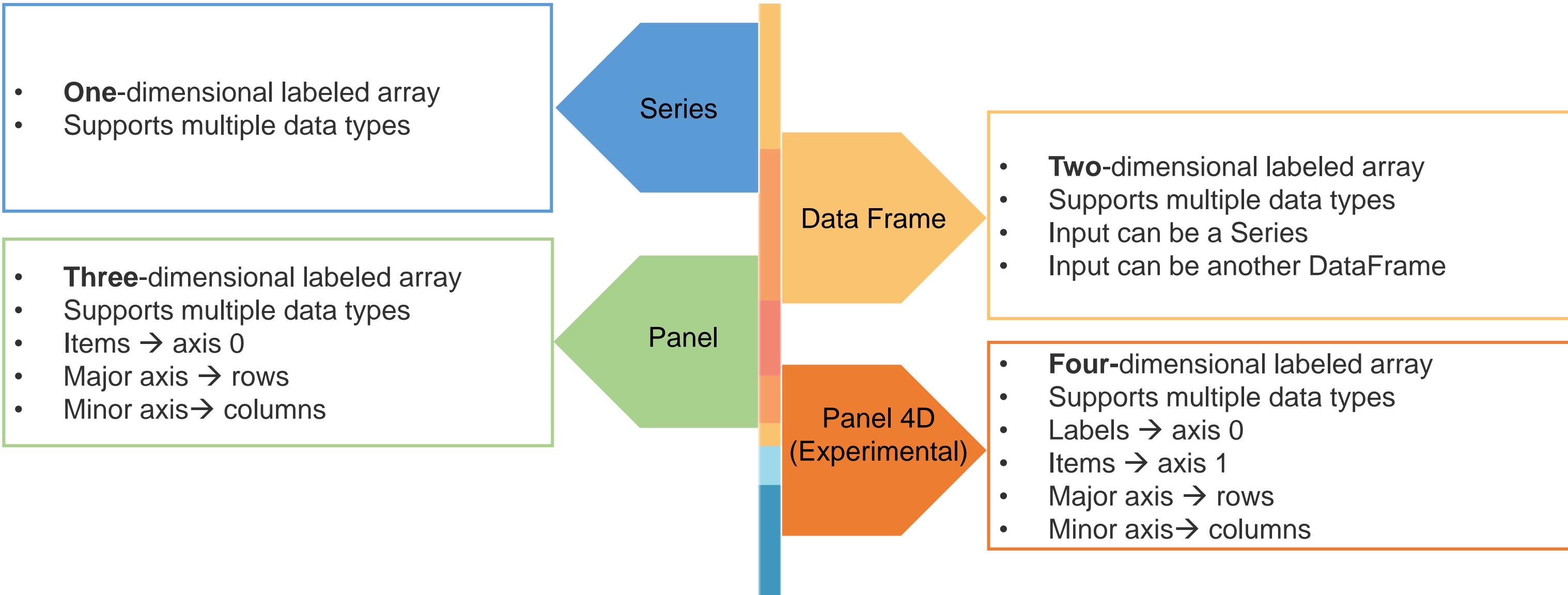
The various features of Pandas makes it an efficient library for Data Scientists.



# Data Structures

---

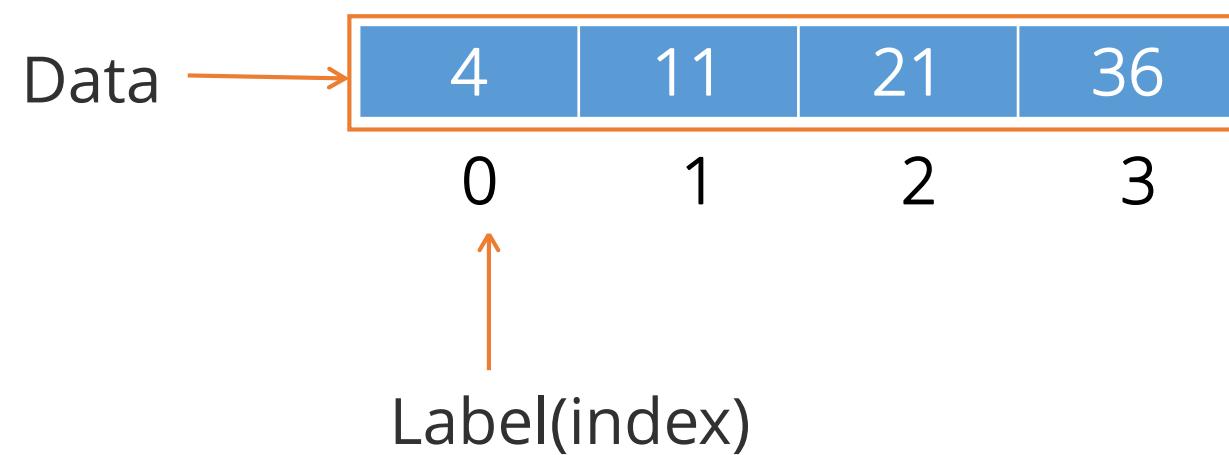
The four main libraries of Pandas data structure are:



# Understanding Series

---

Series is a one-dimensional array-like object containing data and labels (or index).

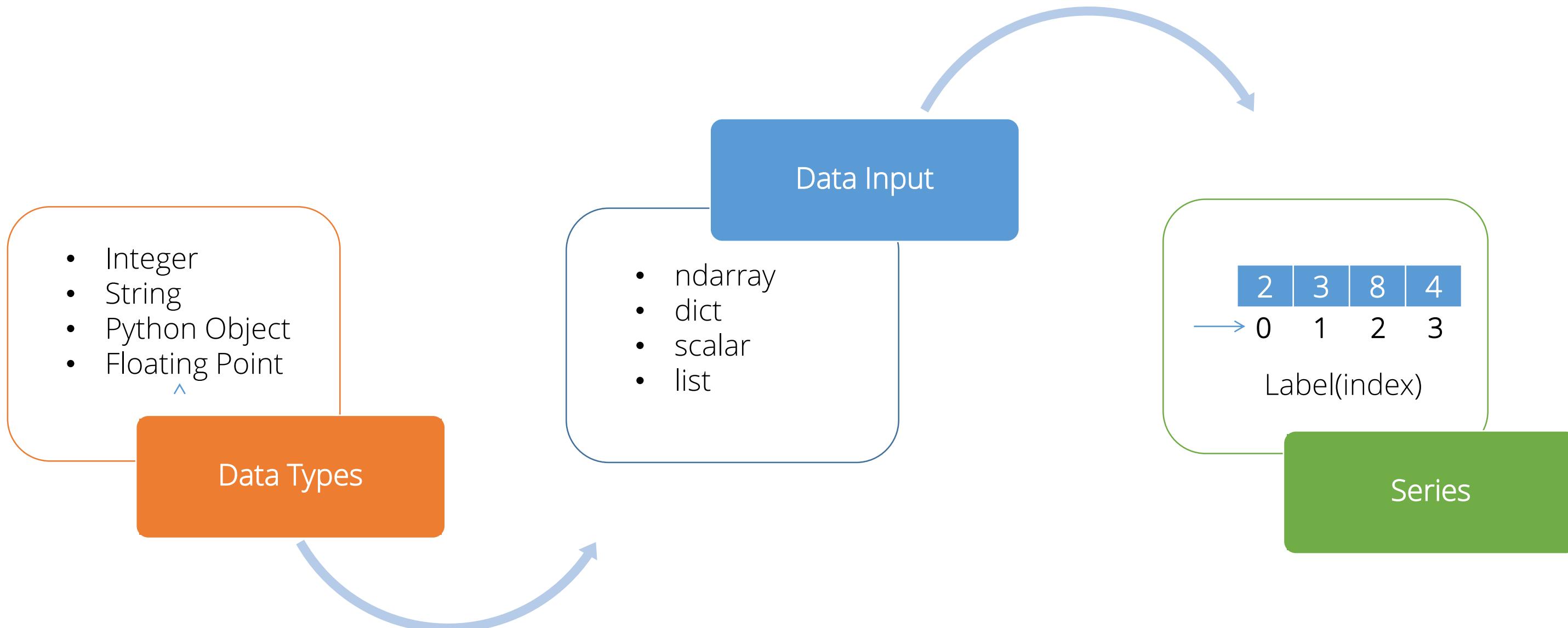


Data alignment is intrinsic and will not be broken until changed explicitly by program.

# Series



Series can be created with different data inputs:

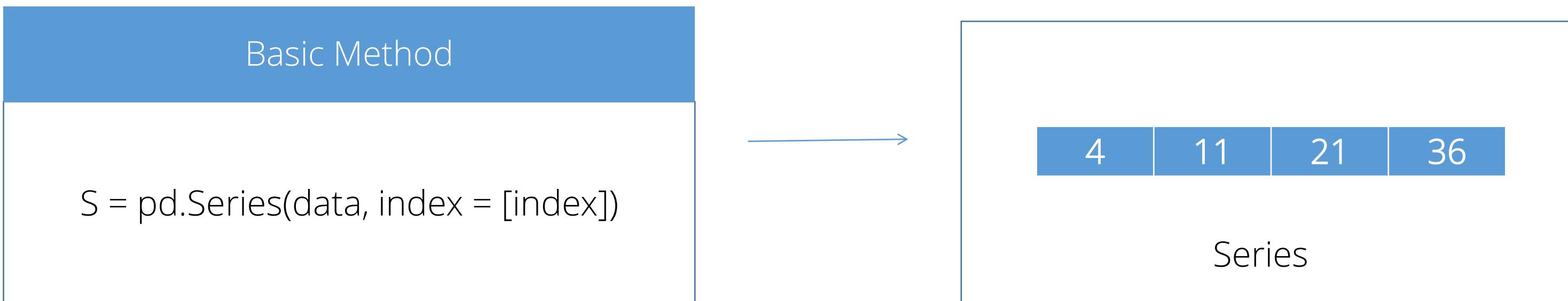


# How to Create Series

---

Key points to note while creating a series are as follows:

- Import Pandas as it is the main library
- Apply the syntax and pass the data elements as arguments
- Import NumPy while working with ndarrays



# Create Series from List

This example shows you how to create a series from a list:

```
In [14]: import numpy as np  
         import pandas as pd
```

Import libraries

```
In [15]: first_series = pd.Series(list('abcdef'))
```

Pass list as an argument

```
In [16]: print(first_series)
```

Index	Data value	Data type
0	a	
1	b	
2	c	
3	d	
4	e	
5	f	
		dtype: object



We have not created index for data but notice that data alignment is done automatically

# Create Series from ndarray

This example shows you how to create a series from an ndarray:

```
In [17]: np_country = np.array(['Luxembourg', 'Norway', 'Japan', 'Switzerland', 'United States', 'Qatar', 'Iceland', 'Sweden',  
                           'Singapore', 'Denmark'])
```

```
In [18]: s_country = pd.Series(np_country) ← Pass ndarray as an argument
```

```
In [19]: print (s_country)
```

0	Luxembourg
1	Norway
2	Japan
3	Switzerland
4	United States
5	Qatar
6	Iceland
7	Sweden
8	Singapore
9	Denmark
	<b>dtype: object</b>

countries ← Data type

# Create Series from dict

A series can also be created with dict data input for faster operations.

dict for countries and their gdp

```
In [10]: #Evaluate countries and their corresponding gdp per capita and print them as series  
dict_country_gdp = pd.Series([52056.01781,40258.80862,40034.85063,39578.07441,39170.41371,37958.23146,37691.02733,  
36152.66676,34706.19047,33630.24604,33529.83052,30860.12808],index=['Luxembourg','Macao, China','Norway',  
'Japan','Switzerland','Hong Kong, China','United States','Qatar','Iceland','Sweden','Singapore','Denmark'])
```

```
In [11]: print (dict_country_gdp)
```

Country	GDP
Luxembourg	52056.01781
Macao, China	40258.80862
Norway	40034.85063
Japan	39578.07441
Switzerland	39170.41371
Hong Kong, China	37958.23146
United States	37691.02733
Qatar	36152.66676
Iceland	34706.19047
Sweden	33630.24604
Singapore	33529.83052
Denmark	30860.12808
	dtype: float64

Countries have been passed as an index  
and GDP as the actual data value

# Create Series from Scalar

```
In [31]: #Print Series with scalar input  
scalar_series = pd.Series(5., index=['a','b','c','d','e'])
```

```
In [32]: scalar_series
```

```
Out[32]:
```

a	5
b	5
c	5
d	5
e	5
dtype: float64	

index

Data type

Data

Scalar input

Index

# Accessing Elements in Series

Data can be accessed through different functions like loc, iloc by passing data element position or index range.

```
In [43]: #access elements in the series  
dict_country_gdp[0]
```

```
Out[43]: 52056.017809999998
```

```
In [44]: #access first 5 countries from the series  
dict_country_gdp[0:5]
```

```
Out[44]: Luxembourg      52056.01781  
Macao, China      40258.80862  
Norway          40034.85063  
Japan            39578.07441  
Switzerland     39170.41371  
dtype: float64
```

```
In [45]: #Look up a country by name or index  
dict_country_gdp.loc['United States']
```

```
Out[45]: 37691.027329999997
```

```
In [46]: #Look up by position  
dict_country_gdp.iloc[0]
```

```
Out[46]: 52056.017809999998
```

# Vectorized Operations in Series

Vectorized operations are performed by the data element's position.

```
In [52]: first_vector_series = pd.Series([1,2,3,4],index=['a','b','c','d'])  
second_vector_series = pd.Series([10,20,30,40],index=['a','b','c','d'])
```

Add the series



```
In [53]: first_vector_series+second_vector_series
```

```
Out[53]: a    11  
          b    22  
          c    33  
          d    44  
          dtype: int64
```



```
In [54]: second_vector_series = pd.Series([10,20,30,40],index=['a','d','b','c'])
```

```
In [55]: first_vector_series+second_vector_series
```



```
Out[55]: a    11  
          b    32  
          c    43  
          d    24  
          dtype: int64
```

# Vectorized Operations in Series

```
In [19]: #now replace few indexes with new ones in second vector series  
second_vector_series = pd.Series([10,20,30,40],index=['a','b','e','f'])
```

```
In [20]: first_vector_series+second_vector_series
```

```
Out[20]:  
a    11  
b    22  
c    NaN  
d    NaN  
e    NaN  
f    NaN  
dtype: float64
```





# Knowledge Check

## How is an index for data elements assigned while creating a Pandas series ? Select all that apply.

- a. Created automatically
- b. Needs to be assigned
- c. Once created can not be changed or altered
- d. Index is not applicable as series is one-dimensional



KNOWLEDGE  
CHECK

**How is an index for data elements assigned while creating a Pandas series ? Select all that apply.**

- a. Created automatically
- b. Needs to be assigned
- c. Once created can not be changed or altered
- d. Index is not applicable as series is one-dimensional



The correct answer is **a, b** .

**Explanation:** Data alignment is intrinsic in Pandas data structure and happens automatically. One can also assign index to data elements.

KNOWLEDGE  
CHECK**What will the result be in vector addition if label is not found in a series?**

- a. Marked as Zeros for missing labels
- b. Labels will be skipped
- c. Marked as NaN for missing labels
- d. Will throw an exception, index not found



KNOWLEDGE  
CHECK

## What will the result be in vector addition if label is not found in a series?

- a. Marked as Zeros for missing labels
- b. Labels will be skipped
- c. Marked as NaN for missing labels
- d. Will throw an exception, index not found



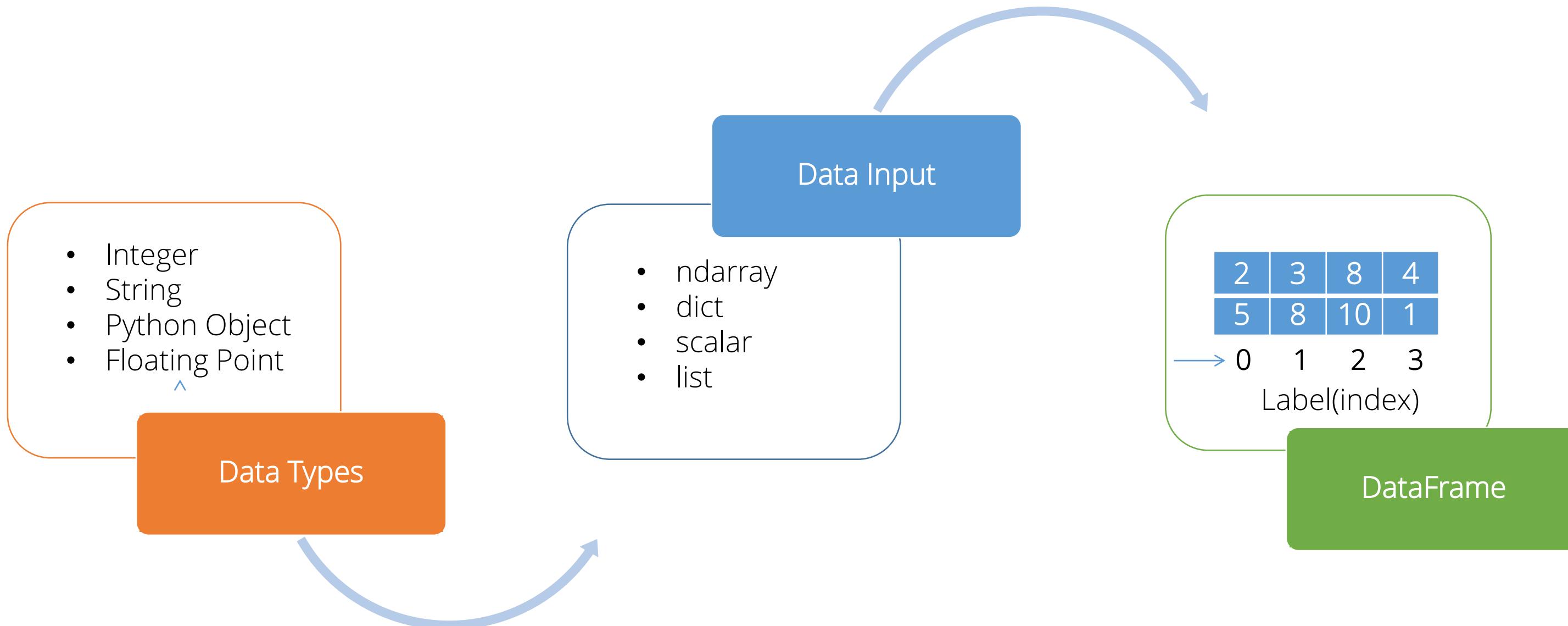
The correct answer is **c** .

**Explanation:** The result will be marked as NaN (Not a Number) for missing labels.

# DataFrame

---

**DataFrame** is a two-dimensional labeled data structure with columns of potentially different types.



# Create DataFrame from Lists

Let's see how you can create a DataFrame from lists:

```
In [1]: import pandas as pd
```

## Create DataFrame from dict of equal length lists

```
In [2]: #Last five olympnics data: place, year and number of countries participated
olympic_data_list = {'HostCity':['London','Beijing','Athens','Sydney','Atlanta'],
                     'Year':[2012,2008,2004,2000,1996],
                     'No. of Participating Countries':[205,204,201,200,197]}
```

```
In [3]: df_olympic_data = pd.DataFrame(olympic_data_list) ← Pass the list to the DataFrame
```

```
In [4]: df_olympic_data
```

Out[4]:

	HostCity	No. of Participating Countries	Year
0	London	205	2012
1	Beijing	204	2008
2	Athens	201	2004
3	Sydney	200	2000
4	Atlanta	197	1996

# Create DataFrame from dict

This example shows you how to create a DataFrame from a series of dicts:

## Create DataFrame from dict of dicts

```
In [5]: olympic_data_dict = {'London':{2012:205}, 'Beijing':{2008:204}}
```

dict one

dict two

```
In [6]: df_olympic_data_dict = pd.DataFrame(olympic_data_dict)
```

Entire dict

```
In [7]: df_olympic_data_dict
```

Out[7]:

	Beijing	London
2008	204	NaN
2012	NaN	205

# View DataFrame

You can view a DataFrame by referring the column name or with the describe function.

```
In [8]: #select by City name  
df_olympic_data.HostCity
```

```
Out[8]: 0    London  
1    Beijing  
2    Athens  
3    Sydney  
4    Atlanta  
Name: HostCity, dtype: object
```

```
In [9]: #use describe function to display the content  
df_olympic_data.describe
```

```
Out[9]: <bound method DataFrame.describe of   HostCity  No. of Participating Countries  Year  
0    London                      205      2012  
1    Beijing                     204      2008  
2    Athens                      201      2004  
3    Sydney                      200      2000  
4    Atlanta                     197      1996>
```

# Create DataFrame from dict of Series

## Create DataFrame from dict of series

```
In [10]: olympic_series_participation = pd.Series([205,204,201,200,197],index=[2012,2008,2004,2000,1996])
olympic_series_country = pd.Series(['London','Beijing','Athens','Sydney','Atlanta'],
                                   index=[2012,2008,2004,2000,1996])
```

```
In [11]: df_olympic_series = pd.DataFrame({'No. of Participating Countries':olympic_series_participation,
                                         'Host Cities':olympic_series_country})
```

```
In [12]: df_olympic_series
```

Out[12]:

	Host Cities	No. of Participating Countries
2012	London	205
2008	Beijing	204
2004	Athens	201
2000	Sydney	200
1996	Atlanta	197

# Create DataFrame from ndarray

## Create DataFrame from dict of ndarray

```
In [13]: import numpy as np
```

```
In [14]: np_array = np.array([2012, 2008, 2004, 2006]) ← Create an ndarray with years  
dict_ndarray = {'year':np_array} ← Create a dict with the ndarray
```

```
In [15]: df_ndarray = pd.DataFrame(dict_ndarray) ← Pass this dict to a new DataFrame
```

```
In [16]: df_ndarray
```

Out[16]:

	year
0	2012
1	2008
2	2004
3	2006

# Create DataFrame from DataFrame

## Create DataFrame from DataFrame object

```
In [17]: df_from_df = pd.DataFrame(df_olympic_series)
```

Create a DataFrame from a DataFrame

```
In [18]: df_from_df
```

Out[18]:

	Host Cities	No. of Participating Countries
2012	London	205
2008	Beijing	204
2004	Athens	201
2000	Sydney	200
1996	Atlanta	197



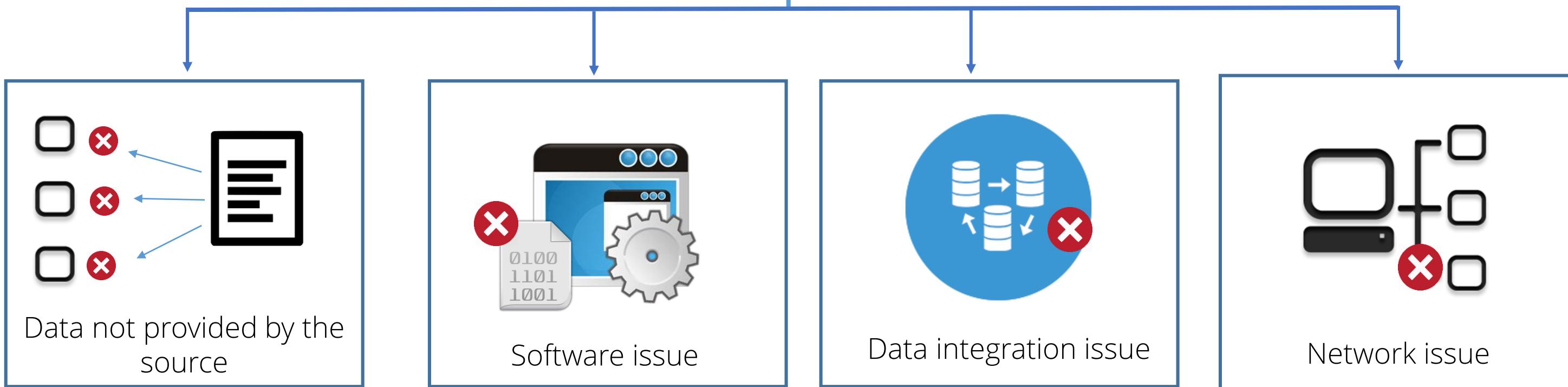
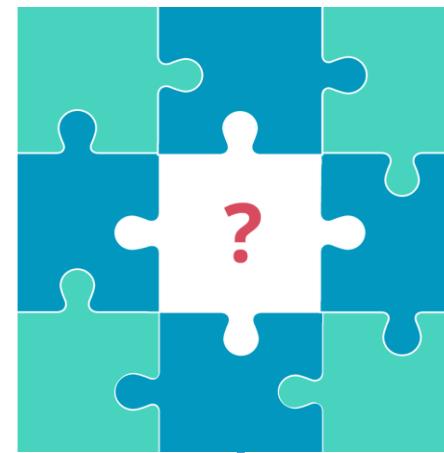
## Demo 01—View and Select Data

Demonstrate how to view and select data in a DataFrame.

DATA  
SCIENCE

# Missing Values

Various factors may lead to missing data values:



# Handle Missing Values

It's difficult to operate on a dataset when it has missing values or uncommon indices.

```
In [3]: import pandas as pd
```

```
In [4]: #declare first series  
first_series = pd.Series([1,2,3,4,5],index=['a','b','c','d','e'])
```

```
In [5]: #declare second series  
second_series=pd.Series([10,20,30,40,50],index=['c','e','f','g','h'])
```

```
In [6]: sum_of_series = first_series+second_series
```

```
In [7]: sum_of_series
```

```
Out[7]:  
a    NaN  
b    NaN  
c    13  
d    NaN  
e    25  
f    NaN  
g    NaN  
h    NaN  
dtype: float64
```

# Handle Missing Values with Functions

The dropna function drops all the values with uncommon indices.

In [5]: sum\_of\_series

Out[5]:

a	NaN
b	NaN
c	13.0
d	NaN
e	25.0
f	NaN
g	NaN
h	NaN
dtype:	float64

In [6]: # drop NaN( Not a Number) values from dataset  
dropna\_s = sum\_of\_series.dropna() ←

In [7]: dropna\_s

Out[7]:

c	13.0
e	25.0
dtype:	float64

# Handle Missing Values with Functions

The fillna function fills all the uncommon indices with a number instead of dropping them.

```
In [8]: dropna_s.fillna(0) ← Fill the missing values with zero
```

```
Out[8]: c    13.0
         e    25.0
         dtype: float64
```

```
In [9]: # Fill NaN( Not a Number) values with Zeroes (0)
         fillna_s = sum_of_series.fillna(0) ←
```

```
In [10]: fillna_s
```

```
Out[10]: a    0.0
          b    0.0
          c    13.0
          d    0.0
          e    25.0
          f    0.0
          g    0.0
          h    0.0
          dtype: float64
```

# Handle Missing Values with Functions- Example

```
In [10]: #fill values with zeroes before performing addition operation for missing indices  
fill_NaN_with_zeros_before_sum =first_series.add(second_series,fill_value=0) ←
```

```
In [11]: fill_NaN_with_zeros_before_sum ←
```

```
Out[11]: a    1  
         b    2  
         c   13  
         d    4  
         e   25  
         f   30  
         g   40  
         h   50  
        dtype: float64
```

# Data Operation

Data operation can be performed through various built-in methods for faster data processing.

```
In [1]: import pandas as pd
```

```
In [2]: #declare movie rating dataframe: ratings from 1 to 5 (star * rating)
df_movie_rating = pd.DataFrame(
    {'movie 1': [5,4,3,3,2,1],
     'movie 2': [4,5,2,3,4,2]},
    index=['Tom','Jeff','Peter','Ram','Ted','Paul']
)
```

```
In [3]: df_movie_rating
```

Out[3]:

	movie 1	movie 2
Tom	5	4
Jeff	4	5
Peter	3	2
Ram	3	3
Ted	2	4
Paul	1	2

# Data Operation with Functions

While performing data operation, custom functions can be applied with the `applymap` method.

```
In [4]: def movie_grade(rating):
    if rating==5:
        return 'A'
    if rating==4:
        return 'B'
    if rating==3:
        return 'C'
    else:
        return 'F'
```

← Declare a custom function

```
In [5]: print movie_grade(5)
```

← Test the function

```
In [6]: df_movie_rating.applymap(movie_grade)
```

← Apply the function to the DataFrame

Out[6]:

	movie 1	movie 2
Tom	A	B
Jeff	B	A
Peter	C	F
Ram	C	C
Ted	F	B
Paul	F	F

# Data Operation with Statistical Functions

This example shows data operations with different statistical functions.

```
In [7]: df_test_scores = pd.DataFrame(  
    {'Test1': [95, 84, 73, 88, 82, 61],  
     'Test2': [74, 85, 82, 73, 77, 79]},  
    index=['Jack', 'Lewis', 'Patrick', 'Rich', 'Kelly', 'Paula'])
```

Create a DataFrame with two test scores for six students.

```
In [8]: df_test_scores.max()
```

Apply the max function to find the maximum score.

```
Out[8]: Test1    95  
Test2    85  
dtype: int64
```

```
In [9]: df_test_scores.mean()
```

Apply the mean function to find the average score.

```
Out[9]: Test1    80.500000  
Test2    78.333333  
dtype: float64
```

```
In [10]: df_test_scores.std()
```

Apply the std function to find the standard deviation for both the tests.

```
Out[10]: Test1    11.979149  
Test2    4.633213  
dtype: float64
```

# Data Operation Using Groupby

This example shows how to operate data using the groupby function.

```
In [16]: df_president_name = pd.DataFrame({'first':['George','Bill', 'Ronald','Jimmy','George'],
                                         'last':['Bush','Clinton', 'Regan', 'Carter', 'Washington']})
```

```
In [17]: df_president_name
```

Out[17]:

	first	last
0	George	Bush
1	Bill	Clinton
2	Ronald	Regan
3	Jimmy	Carter
4	George	Washington

Create a DataFrame with first and last name as former presidents

```
In [18]: grouped = df_president_name.groupby('first')
```

Group the DataFrame with the first name

```
In [19]: grp_data = grouped.get_group('George')
grp_data
```

Group the DataFrame with the first name

Out[19]:

	first	last
0	George	Bush
4	George	Washington

# Data Operation – Sorting

This example shows how to sort data

In [20]: `df_president_name.sort_values('first')` ← Sort values by first name

Out[20]:

	first	last
1	Bill	Clinton
0	George	Bush
4	George	Washington
3	Jimmy	Carter
2	Ronald	Regan



## Demo 02—Data Operations

Demonstrate how to perform data operations.

DATA  
SCIENCE

# Data Standardization

This example shows how to standardize a dataset.

```
In [11]: def standardize_tests(test):
    return (test-test.mean())/ test.std() ← Create a function to return the standardize value
```

```
In [12]: standardize_tests(df_test_scores['Test1'])
```

```
Out[12]: Jack      1.210437
          Lewis     0.292174
          Patrick   -0.626088
          Rich      0.626088
          Kelly     0.125218
          Paula    -1.627829
          Name: Test1, dtype: float64
```

```
In [13]: def standardize_test_scores(datafrm):
    return datafrm.apply(standardize_tests) ← Apply the function to the entire dataset
```

```
In [14]: standardize_test_scores(df_test_scores)
```

```
Out[14]:
```

	Test1	Test2
Jack	1.210437	-0.935276
Lewis	0.292174	1.438886
Patrick	-0.626088	0.791387
Rich	0.626088	-1.151109
Kelly	0.125218	-0.287777
Paula	-1.627829	0.143889

Standardized test data is applied for the entire DataFrame



# Knowledge Check

KNOWLEDGE  
CHECK**What is the result of DataFrame[3:9]?**

- a. Series with sliced index from 3 to 9
- b. dict of index position 3 and index position 9
- c. DataFrame of sliced rows index from 3 to 9
- d. DataFrame with data elements at index 3 to index9



KNOWLEDGE  
CHECK**What is the result of DataFrame[3:9]?**

- a. Series with sliced index from 3 to 9
- b. dict of index position 3 and index position 9
- c. DataFrame of sliced rows index from 3 to 9
- d. DataFrame with data elements at index 3 to index9



The correct answer is

. c

Explanation: This is DataFrame slicing technique with indexing or selection on data elements. When a user passes the range 3:9, the entire range from 3 to 9 gets sliced and displayed as output.

KNOWLEDGE  
CHECK**What does the fillna() method do?**

- a. Fills all NaN values with zeros
- b. Fills all NaN values with one
- c. Fills all NaN values with values mentioned in the parenthesis
- d. Drops NaN values from the dataset



KNOWLEDGE  
CHECK**What does the fillna() method do?**

- a. Fills all NaN values with zeros
- b. Fills all NaN values with One
- c. Fills all NaN values with values mentioned in the parenthesis
- d. Drops NaN values from the dataset

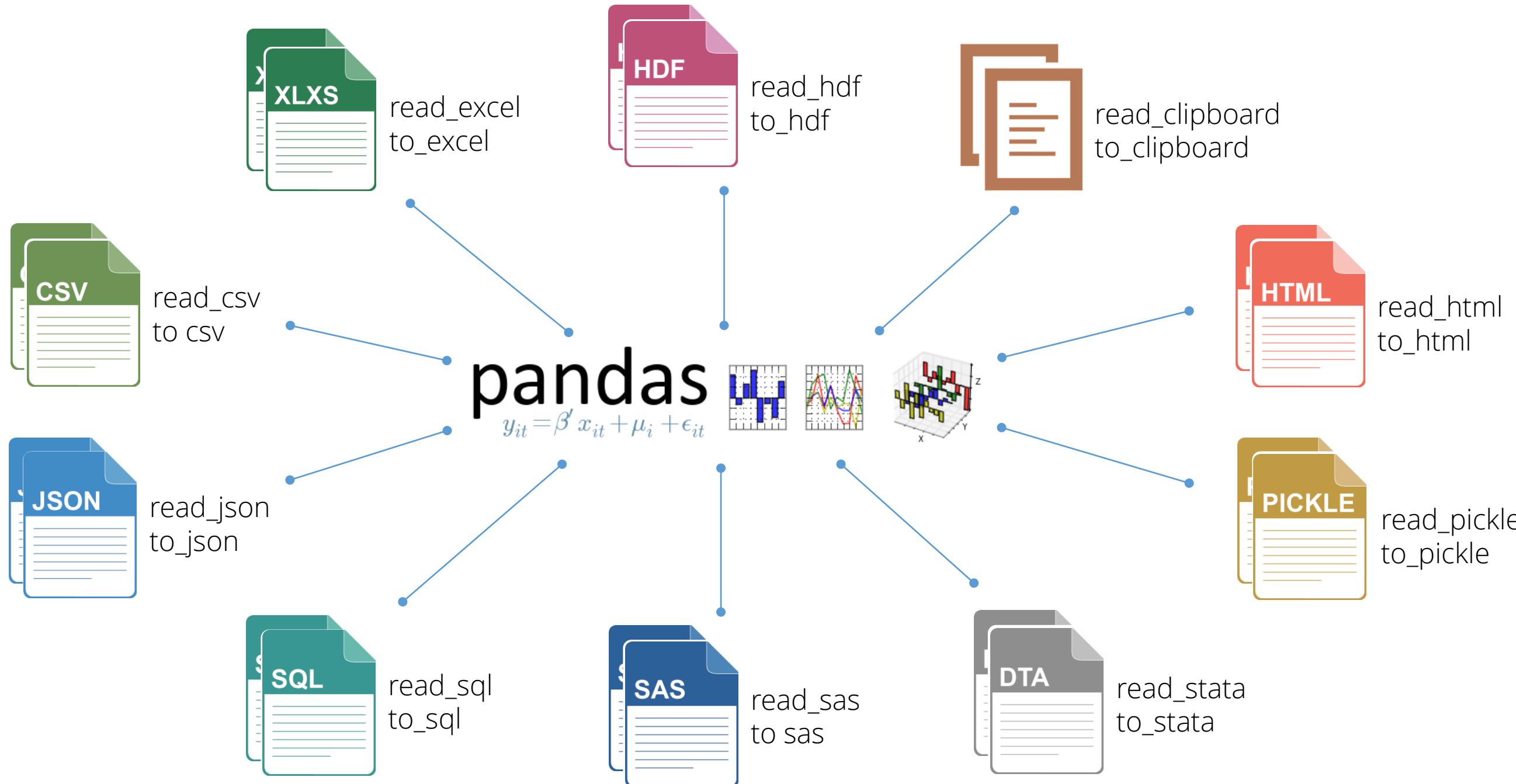


The correct answer is

. c

Explanation: fillna is one of the basic methods to fill NaN values in a dataset with a desired value by passing that in parenthesis.

# File Read and Write Support



# Pandas SQL operation

```
In [1]: #import pandas Library  
import pandas as pd
```

```
In [2]: #import sql lite  
import sqlite3
```

```
In [3]: #Create SQL table  
create_table = """  
CREATE TABLE student_score  
(Id INTEGER, Name VARCHAR(20), Math REAL,  
Science REAL  
);"""
```

```
In [4]: #execute the SQL statement  
executeSQL = sqlite3.connect(':memory:')  
executeSQL.execute(create_table)  
executeSQL.commit()
```

```
In [5]: #prepare a SQL query  
SQL_query = executeSQL.execute('select * from student_score')
```

```
In [7]: #fetch result from the SQLlite database  
resulset = SQL_query.fetchall()
```

```
In [8]: #view result (empty data)  
resulset
```

```
Out[8]: []
```

# Pandas SQL operation

```
In [9]: #prepare records to be inserted into SQL table through SQL statement
insertSQL = [(10,'Jack',85,92),
             (29,'Tom',73,89),
             (65,'Ram',65.5,77),
             (5,'Steve',55,91)
            ]
```

```
In [10]: #insert records into SQL table through SQL statement
insert_statement = "Insert into student_score values(?, ?, ?, ?, ?)"
executeSQL.executemany(insert_statement,insertSQL)
executeSQL.commit()
```

```
In [11]: #prepare SQL query
SQL_query = executeSQL.execute("select * from student_score")
```

```
In [12]: #fetch the resultset for the query
resulset = SQL_query.fetchall()
```

```
In [13]: #view the resultset
resulset
```

```
Out[13]: [(10, u'Jack', 85.0, 92.0),
           (29, u'Tom', 73.0, 89.0),
           (65, u'Ram', 65.5, 77.0),
           (5, u'Steve', 55.0, 91.0)]
```

# Pandas SQL operation

In [14]: *#put the records together in dataframe*

```
df_student_recors = pd.DataFrame(resulset,columns=zip(*SQL_query.description)[0])
```

In [15]: *#view the records in pandas dataframe*

```
df_student_recors
```

Out[15]:

	<b>Id</b>	<b>Name</b>	<b>Math</b>	<b>Science</b>
<b>0</b>	10	Jack	85.0	92.0
<b>1</b>	29	Tom	73.0	89.0
<b>2</b>	65	Ram	65.5	77.0
<b>3</b>	5	Steve	55.0	91.0

# Activity—Sequence it Right!

The code here is buggy. You have to correct its sequence to debug it. To do that, click any two code snippets, which you feel are out of place, to swap their places.

1

```
df_movie_rating = pd.DataFrame(  
    {'movie 1': [5,4,3,3,2,1],  
     'movie 2': [4,5,2,3,4,2]},  
    index=['Tom', 'Jeff', 'Peter', 'Ram', 'Ted', 'Paul'])
```

2

```
print movie_grade(5)
```

A

3

```
def movie_grade(rating):  
    if rating==5:  
        return 'A'  
    if rating==4:  
        return 'B'  
    if rating==3:  
        return 'C'  
    else:  
        return 'F'
```

4

```
df_movie_rating.applymap(movie_grade)
```

*Click any two code snippets to swap them.*



# Assignment

Problem

Instructions

Analyze the Federal Aviation Authority (FAA) dataset using Pandas to do the following:

1. View
  - a. Aircraft make name
  - b. State name
  - c. Aircraft model name
  - d. Text information
  - e. Flight phase
  - f. Event description type
  - g. Fatal flag
2. Clean the dataset and replace the fatal flag NaN with "No"
3. Find the aircraft types and their occurrences in the dataset
4. Remove all the observations where aircraft names are not available
5. Display the observations where fatal flag is "Yes"

Problem

Instructions

Instructions to perform the assignment:

- Download the FAA dataset from the “Resource” tab. Upload the dataset to your Jupyter notebook to view and evaluate it.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



# Assignment

Problem

Instructions

A dataset in CSV format is given for the Fire Department of New York City. Analyze the dataset to determine:

1. The total number of fire department facilities in New York city
2. The number of fire department facilities in each borough
3. The facility names in Manhattan

Problem

Instructions

Instructions to perform the assignment:

- Download the FDNY dataset from the “Resource” tab. You can upload the dataset to your Jupyter notebook to use it.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the cues provided to complete the assignment.



**QUIZ**

1

**Which of the following data structures is used to store three-dimensional data?**

- a. Series
- b. DataFrame
- c. Panel
- d. PanelND



**QUIZ  
1**

**Which of the following data structures is used to store three-dimensional data?**

- a. Series
- b. DataFrame
- c. Panel
- d. PanelND



The correct answer is **c**.

**Explanation:** Panel is a data structure used to store three-dimensional data.

**QUIZ  
2**

**Which method is used for label-location indexing by label?**

- a. iat
- b. iloc
- c. loc
- d. std



**QUIZ  
2**

**Which method is used for label-location indexing by label?**

- a. iat
- b. iloc
- c. loc
- d. std



The correct answer is **c**.

**Explanation:** The loc method is used to for label-location indexing by label; iat is strictly integer location and iloc is integer-location-based indexing by position.

**QUIZ  
3**

**While viewing a dataframe, head() method will \_\_\_\_.**

- a. return only the first row
- b. return only headers or column names of the DataFrame
- c. return the first five rows of the DataFrame
- d. throw an exception as it expects parameter(number) in parenthesis



**QUIZ  
3**

**While viewing a dataframe, head() method will \_\_\_\_.**

- a. return only the first row
- b. return only headers or column name of the DataFrame
- c. return the first five rows of the DataFrame
- d. throw an exception as it expects parameter(number) in parenthesis



The correct answer is **c**.

**Explanation:** The default value is 5 if nothing is passed in head method. So it will return the first five rows of the DataFrame.

# Key Takeaways

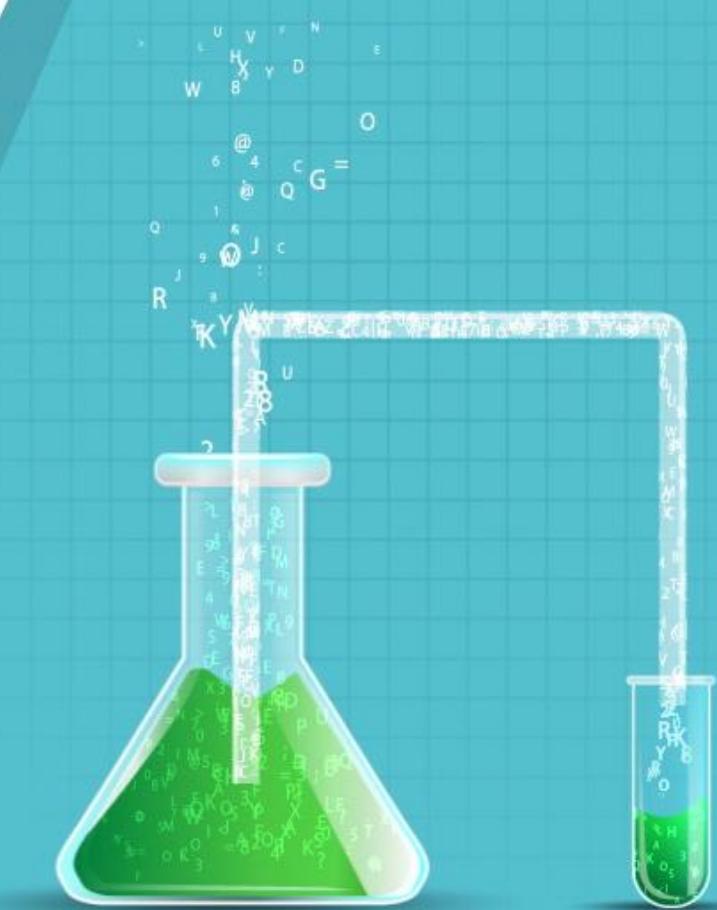
Let us take a quick recap of what we have learned in the lesson:

- Pandas is an open source library for data analysis and is an efficient data wrangling tool in Python.
  - The four main libraries of Pandas are Series, DataFrame, Panel, and Panel 4D.
  - DataFrame is a two-dimensional labeled data structure with columns of potentially different data types.
  - To access data elements in a series, 'loc' and 'iloc' methods can be used.



# Key Takeaways

- The 'iat' method enables selection of elements in a DataFrame by index position and returns the corresponding data element.
  - Missing data values in Pandas can be resolved through two built-in methods such as dropna andfillna.
  - Pandas supports multiple files for data analysis such as Excel, PyTables, Clipboard, HTML, pickle, dta, SAS, SQL, JSON, and CSV.



**This concludes “Data Manipulation with Pandas.”**

The next lesson is “Machine Learning with SciKit Learn.”

DATA  
SCIENCE

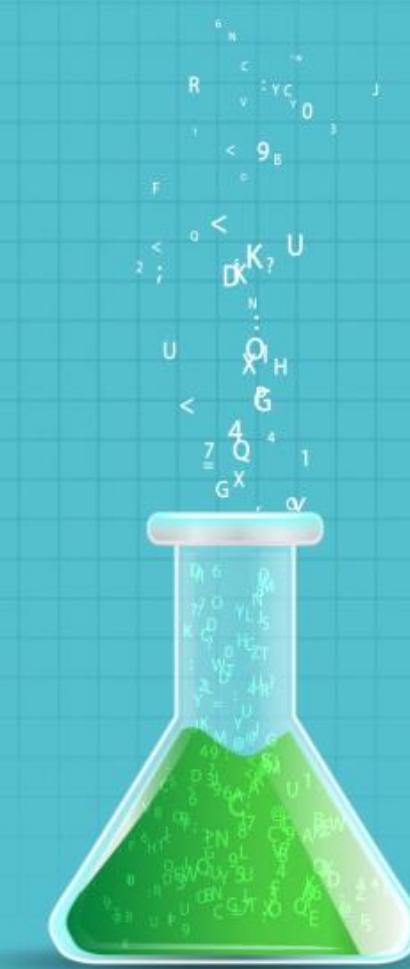
# Data Science with Python

## Lesson 8—Machine Learning with Scikit-Learn



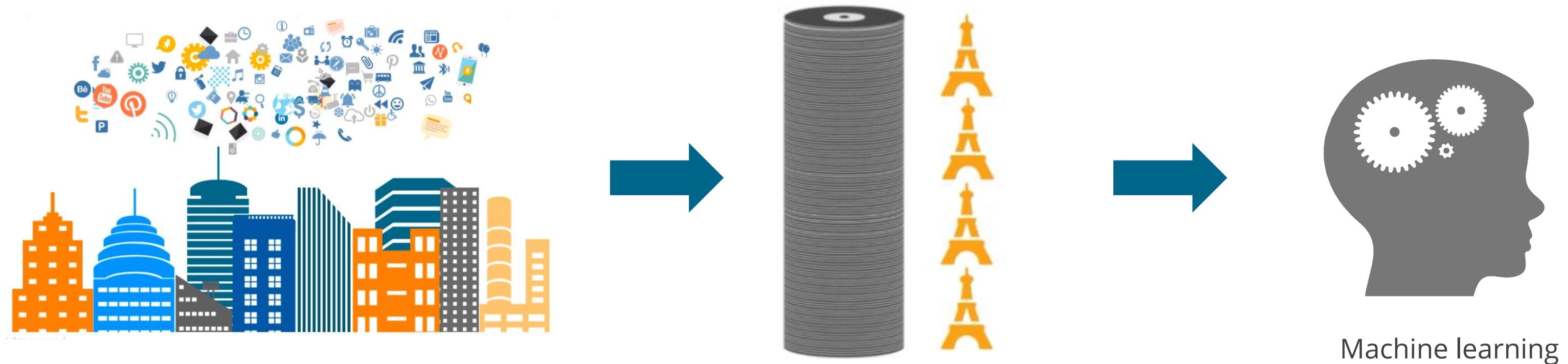
# What You'll Learn

- What machine learning is and why it is important
- The machine learning approach
- Relevant terminologies that help you understand a dataset
- Features of supervised and unsupervised learning models
- Algorithms such as regression, classification, clustering, and dimensionality reduction



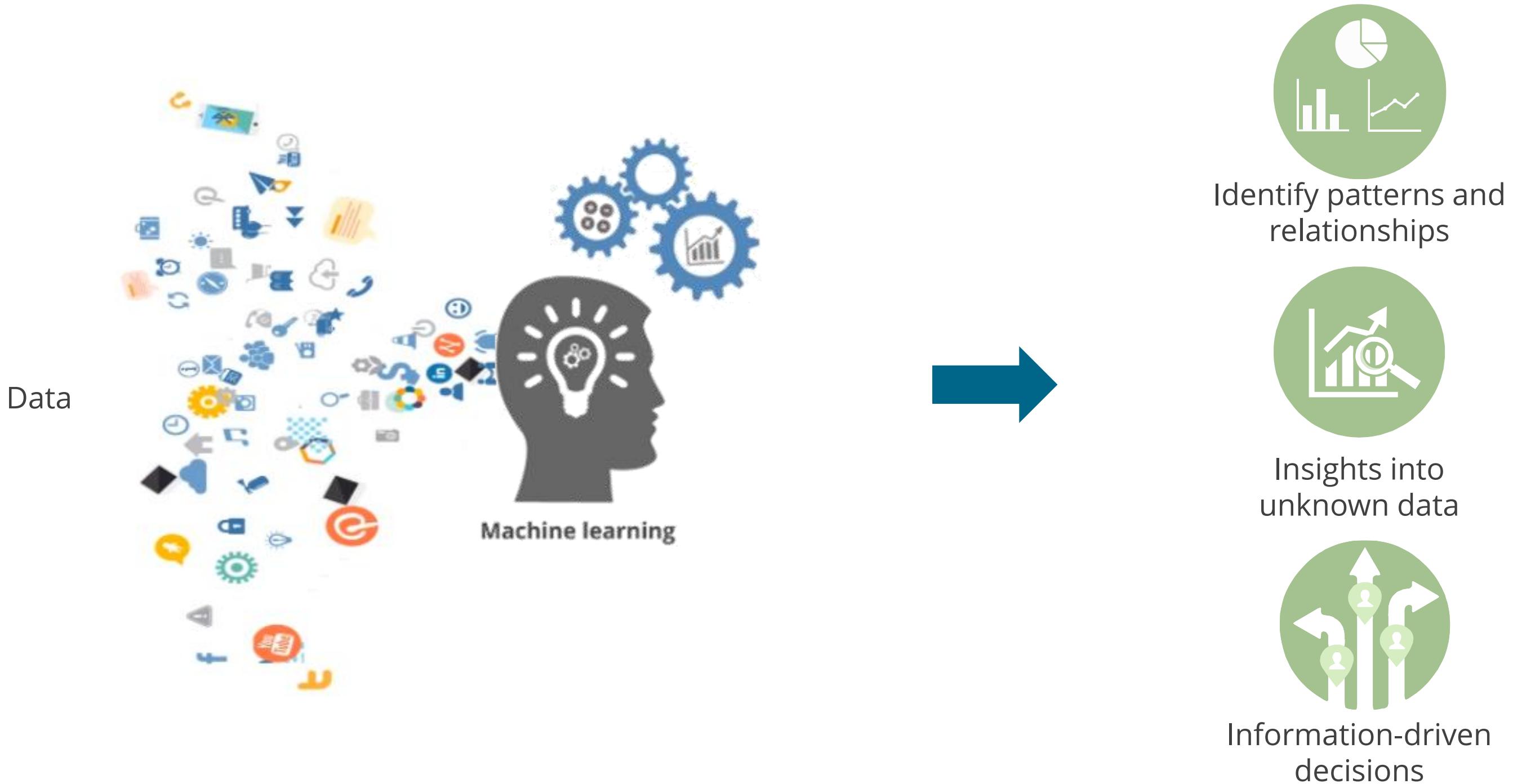
# Why Machine Learning

If we stored the data generated in a day on Blu-ray disks and stacked them up, it would be equal to the height of four Eiffel towers! Machine learning helps analyze this data easily and quickly.



# Purpose of Machine Learning

Machine learning is a great tool to analyze data, find hidden data patterns and relationships, and extract information to enable information-driven decisions and provide insights.



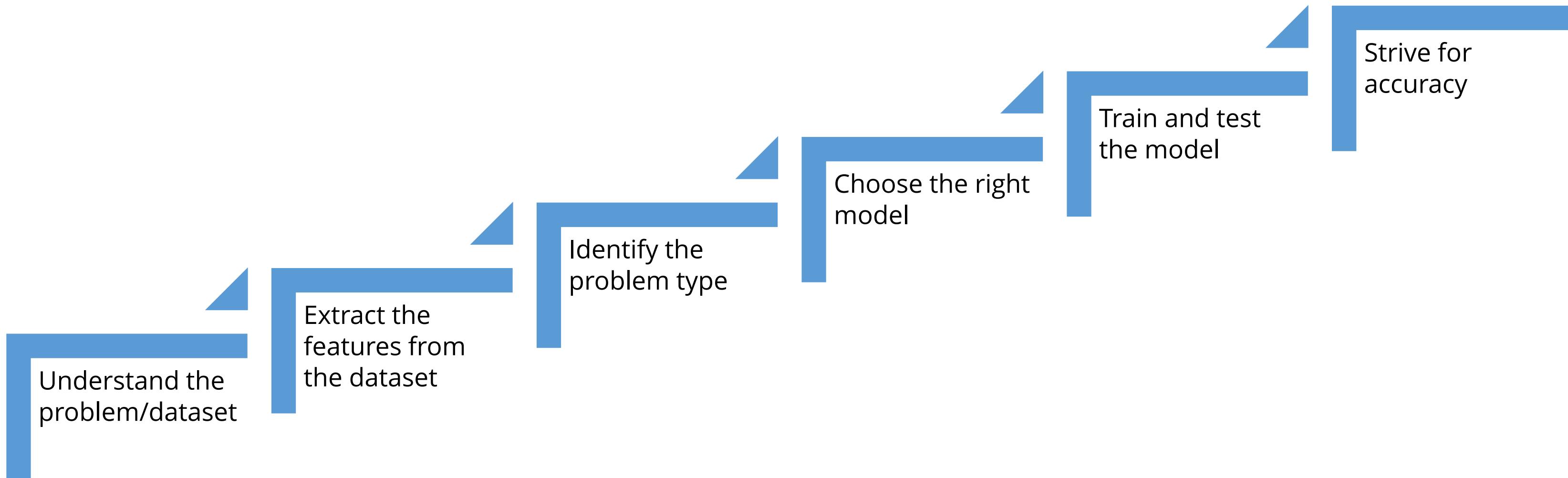
# Machine Learning Terminology

These are some machine learning terminologies that you will come across in this lesson:



# Machine Learning Approach

The machine learning approach starts with either a problem that you need to solve or a given dataset that you need to analyze.



# Steps 1 and 2: Understand the Dataset and Extract its Features

Let us look at a dataset and understand its features in terms of machine learning.

Features (attributes)	Education (Yrs.)	Professional Training (Yes/No)	Hourly Rate (USD)	Response (label)
Observations (records)	16	1	90	
	15	0	65	
	12	1	70	
	18	1	130	
	16	0	110	
	16	1	100	
	15	1	105	
	31	0	70	

Predictors

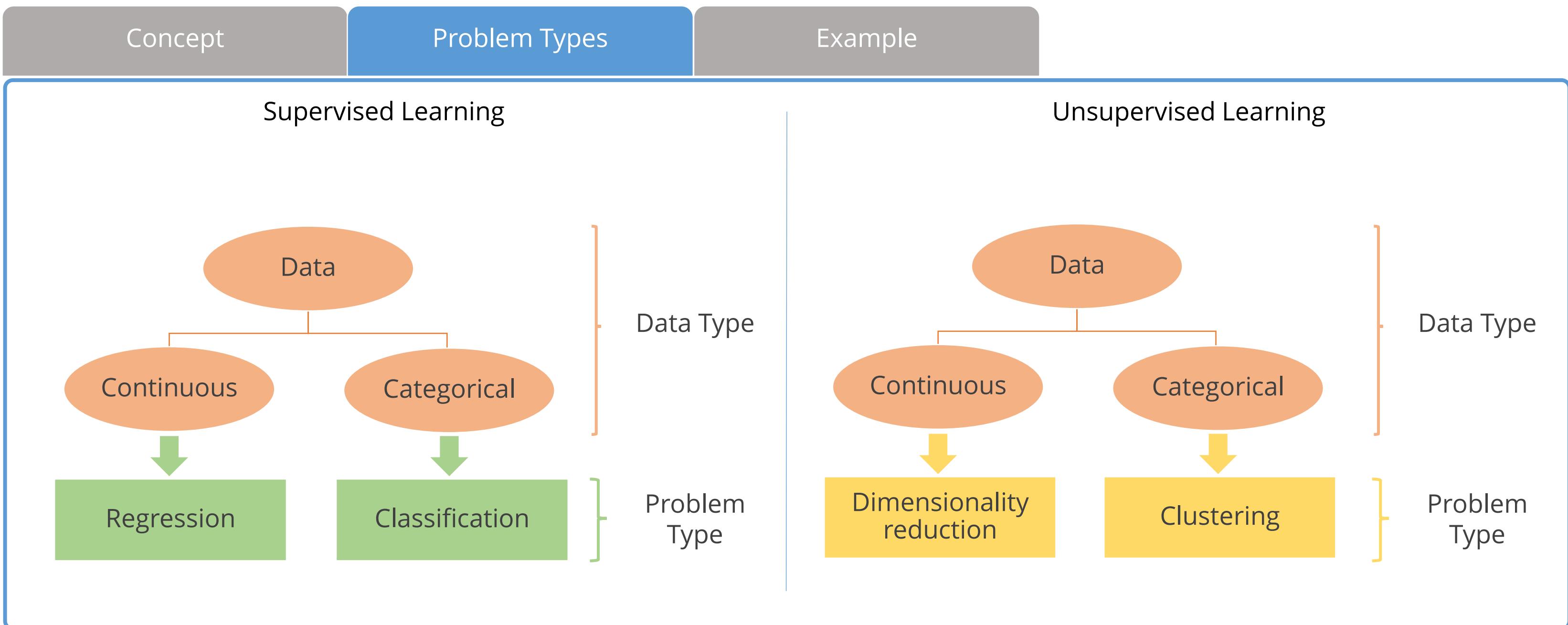
## Steps 3 and 4: Identify the Problem Type and Learning Model

Machine learning can either be supervised or unsupervised. The problem type should be selected based on the type of learning model.

Concept	Problem Types	Example	
Supervised Learning		Unsupervised Learning	
<ul style="list-style-type: none"><li>In supervised learning, the dataset used to train a model should have observations, features, and responses. The model is trained to predict the “right” response for a given set of data points.</li><li>Supervised learning models are used to predict an outcome.</li><li>The goal of this model is to “generalize” a dataset so that the “general rule” can be applied to new data as well.</li></ul>		<ul style="list-style-type: none"><li>In unsupervised learning, the response or the outcome of the data is not known.</li><li>Supervised learning models are used to identify and visualize patterns in data by grouping similar types of data.</li><li>The goal of this model is to “represent” data in a way that meaningful information can be extracted.</li></ul>	

## Steps 3 and 4: Identify the Problem Type and Learning Model (contd.)

Data can either be continuous or categorical. Based on whether it is supervised or unsupervised learning, the problem type will differ.



# Steps 3 and 4: Identify the Problem Type and Learning Model (contd.)

Some examples of supervised and unsupervised learning models are shown here.

Concept

Problem Types

Example

Supervised Learning

The screenshot shows a news aggregator interface. At the top, there are tabs for 'World' and 'U.S.'. Below each tab, there is a list of news stories. The 'World' section includes stories about Jeremy Corbyn, US military presence in the Philippines, Barack Obama's visit to the CIA, and the search for kidnapped Nigerian girls. The 'U.S.' section includes stories about a deputy constable shot in Houston and a couple kissing through a robbery. Each story has a thumbnail image, the title, the source (e.g., BBC News, Reuters), and a timestamp.

Categories of news based on the topics

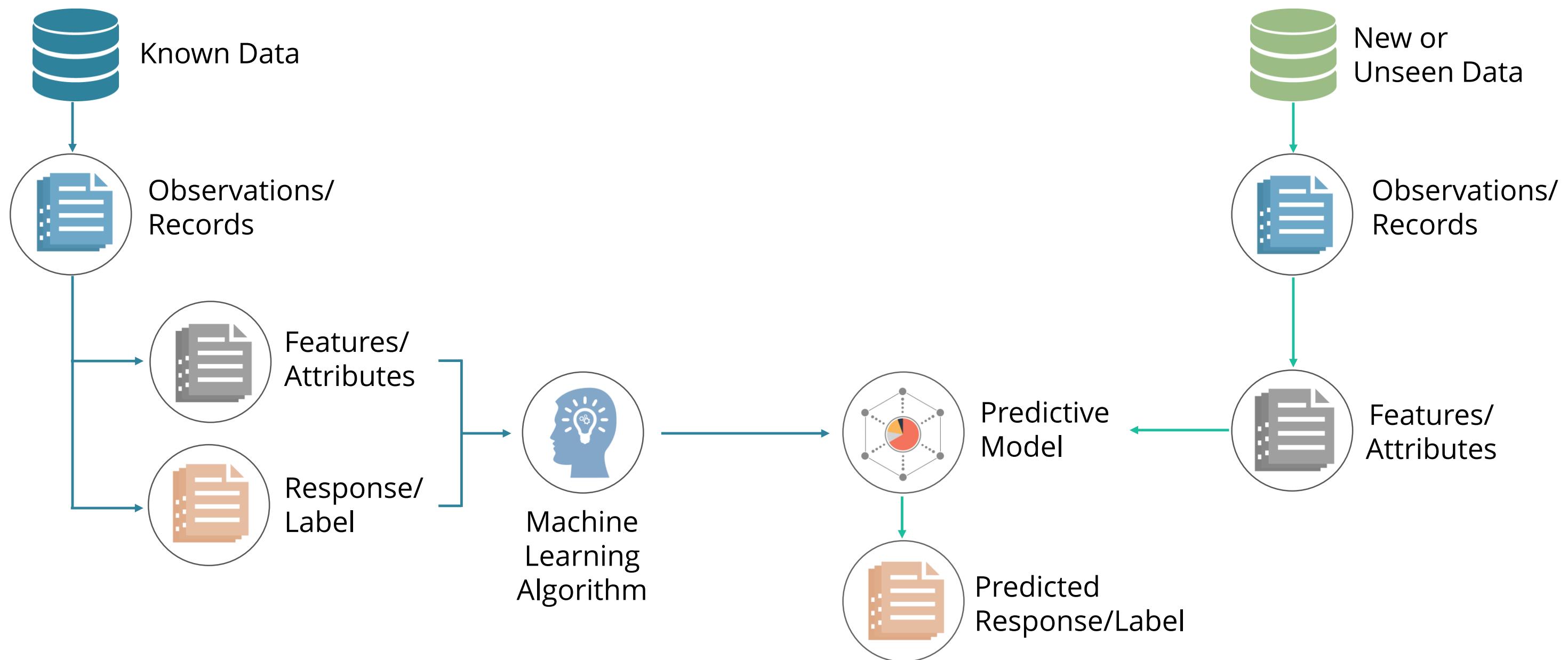
Unsupervised Learning

The screenshot shows a news website's homepage. At the top, there is a search bar and a dropdown menu set to 'U.S. edition'. Below the search bar, there is a 'Top Stories' section featuring a large image of the CNN logo and several smaller news items. One prominent story is titled 'What to Look For in the Democratic Debate' from the New York Times. Below this, there are other stories about Hillary Clinton and the Democratic debate. At the bottom, there is a row of video thumbnails from various news networks (Reuters, Fox News, Fox Business, Los Angeles Times, Washington Post) under the heading 'See realtime coverage'.

Grouping of similar stories on different news networks

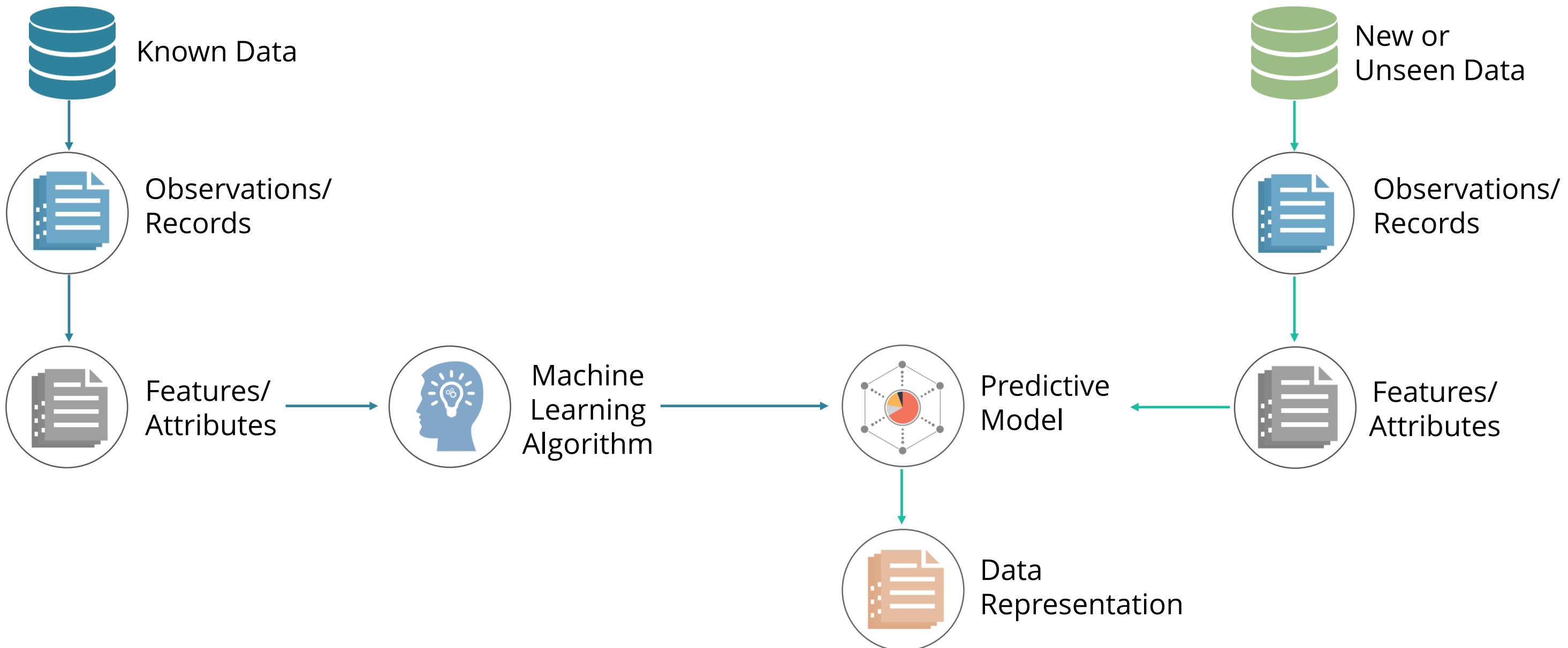
# How it Works—Supervised Learning Model

In supervised learning, a known dataset with observations, features, and response is used to create and train a machine learning algorithm. A predictive model, built on top of this algorithm, is then used to predict the response for a new dataset that has the same features.



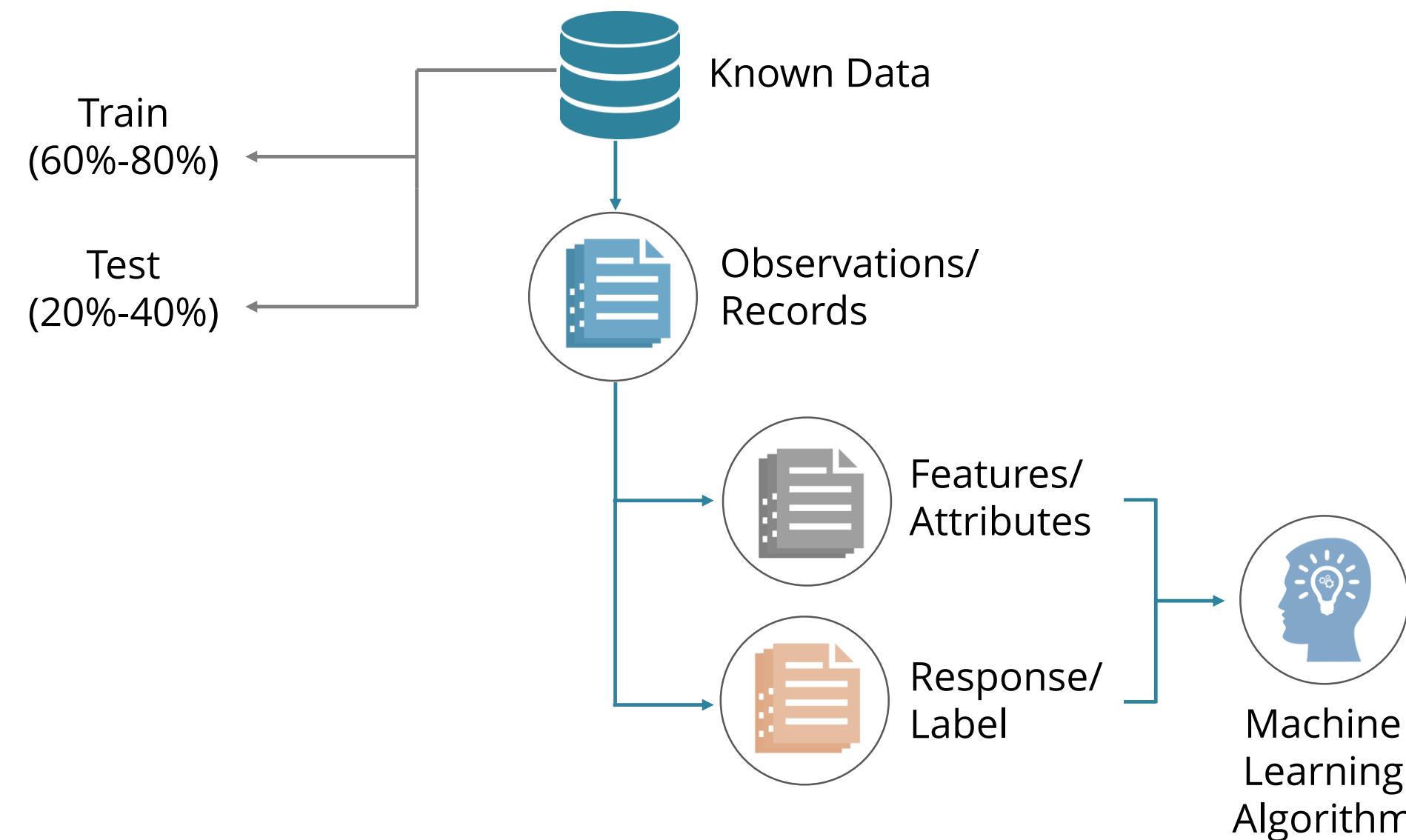
# How it Works—Unsupervised Learning Model

In unsupervised learning, a known dataset has a set of observations with features. But the response is not known. The predictive model uses these features to identify how to classify and represent the data points of new or unseen data.



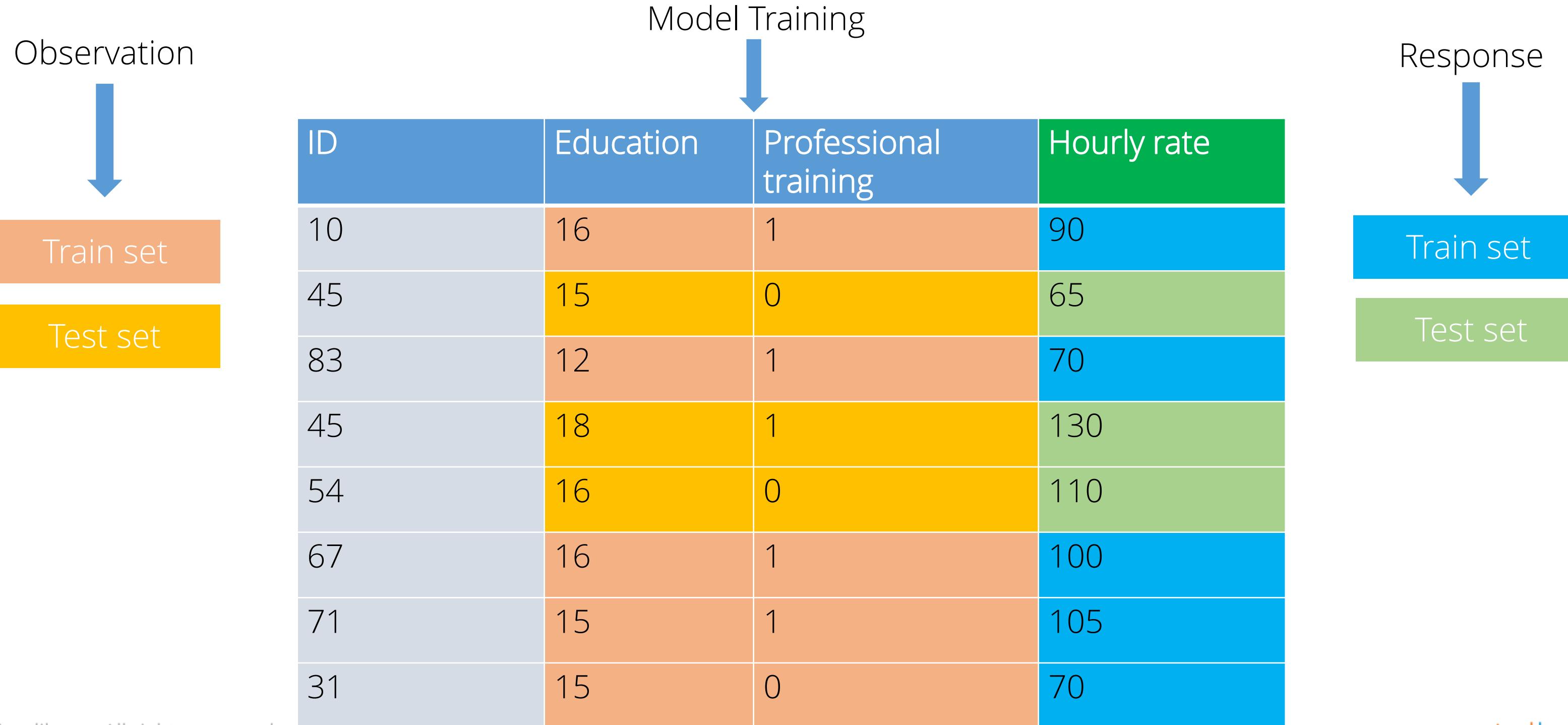
## Steps 5 and 6: Train, Test, and Optimize the Model

To train supervised learning models, data analysts usually divide a known dataset into training and testing sets.



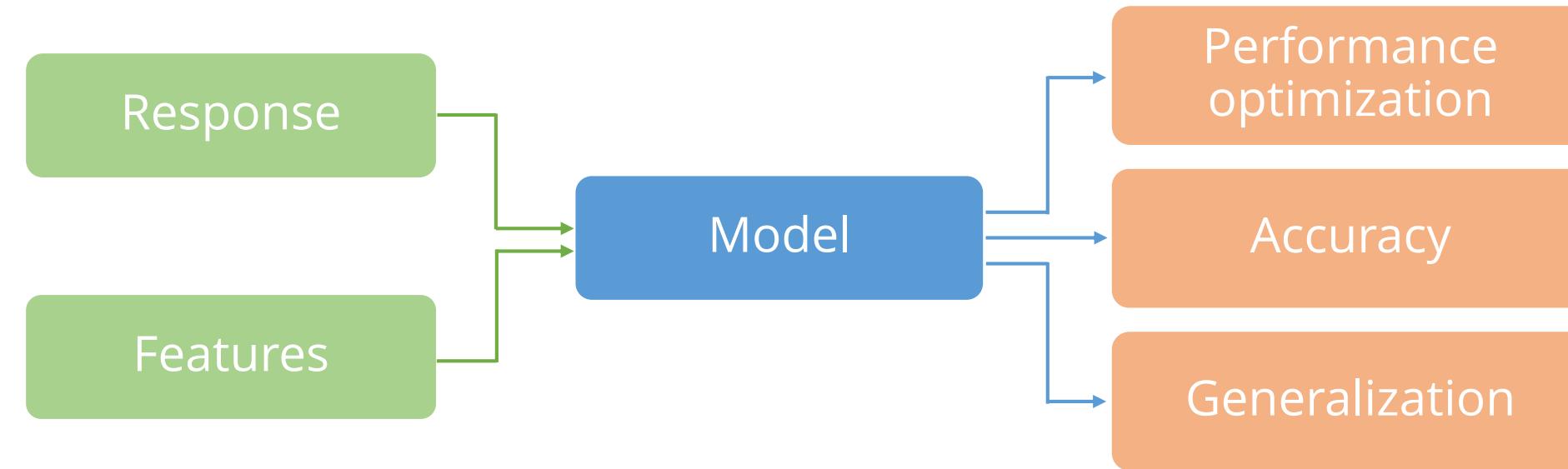
## Steps 5 and 6: Train, Test, and Optimize the Model (contd.)

Let us look at an example to see how the split approach works.



# Supervised Learning Model Considerations

Some considerations of supervised and unsupervised learning models are shown here.





# Knowledge Check

## In machine learning, which one of the following is an observation?

- a. Features
- b. Attributes
- c. Records
- d. Labels



## In machine learning, which one of the following is an observation?

- a. Features
- b. Attributes
- c. Records
- d. Labels



The correct answer is **c**.

**Explanation:** An observation is a set of examples, records, or samples.

**If data is continuous and has labels (response), then it fits which of the following problem types?**

- a. Supervised learning: classification
- b. Unsupervised learning: clustering
- c. Unsupervised learning: dimensionality reduction
- d. Supervised learning: regression



**If data is continuous and has labels (response), then it fits which of the following problem types?**

- a. Supervised learning: classification
- b. Unsupervised learning: clustering
- c. Unsupervised learning: dimensionality reduction
- d. Supervised learning: regression



The correct answer is **d**.

**Explanation:** The regression algorithm belonging to the supervised learning model is best suited to analyze continuous data.

## Identify the goal of unsupervised learning. *Select all that apply.*

- a. To predict the outcome
- b. To understand the structure of the data
- c. To generalize the dataset
- d. To represent the data



## Identify the goal of unsupervised learning. *Select all that apply.*

- a. To predict the outcome
- b. To understand the structure of the data
- c. To generalize the dataset
- d. To represent the data

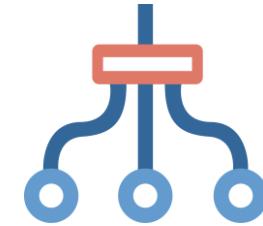


The correct answer is **b, d.**

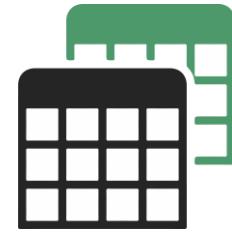
**Explanation:** The goal of unsupervised learning is to understand the structure of the data and represent it. There is no right or certain answer in unsupervised learning.

# Scikit-Learn

Scikit is a powerful and modern machine learning Python library for fully and semi-automated data analysis and information extraction.



Efficient tools to identify  
and organize problems  
(Supervised/ Unsupervised)



Free and open  
datasets



Rich set of libraries  
for learning and  
predicting



Model support for  
every problem type



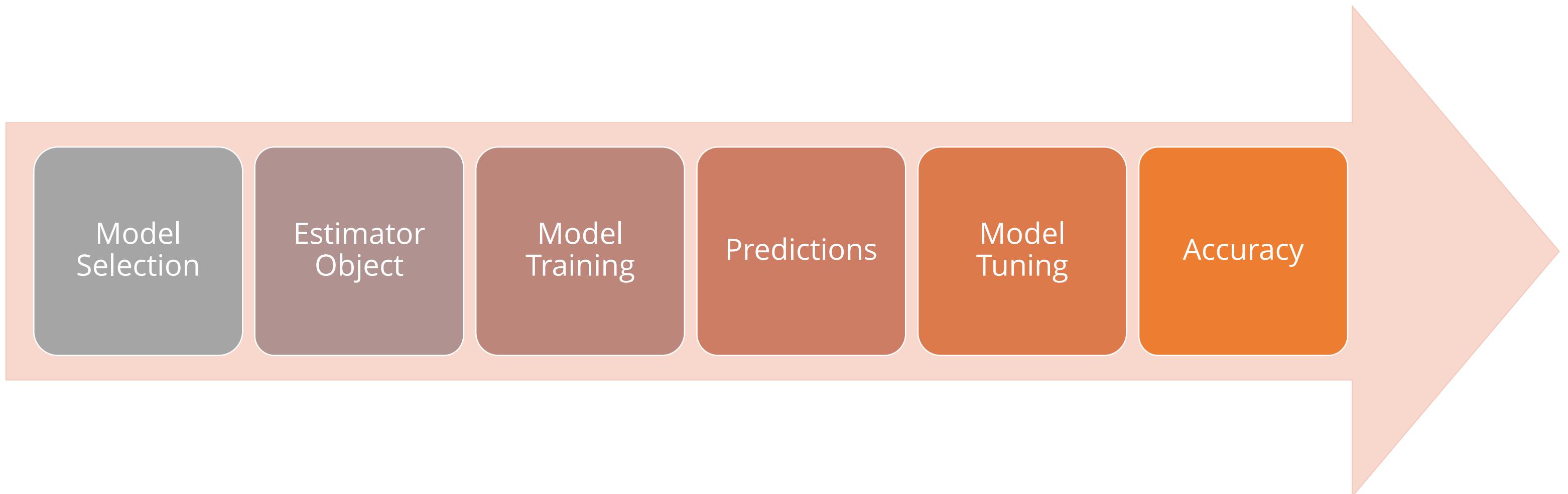
Model  
persistence



Open source  
community and  
vendor support

# Scikit-Learn—Problem-Solution Approach

Scikit-learn helps Data Scientists organize their work through its problem-solution approach.



# Scikit-Learn—Problem-Solution Considerations

While working with a Scikit-Learn dataset or loading your own data to Scikit -Learn, always consider these four points:



Create separate objects for feature and response.



Ensure that features and response have only numeric values.



Features and response should be in the form of a NumPy ndarray.



Since features and response would be in the form of arrays, they would have shapes and sizes.



Features are always mapped as  $x$ , and response is mapped as  $y$ .



# Knowledge Check

**The estimator instance in Scikit-learn is a \_\_\_\_.**

- a. model
- b. feature
- c. dataset
- d. response



**The estimator instance in Scikit-learn is a \_\_\_\_.**

- a. model
- b. feature
- c. dataset
- d. response

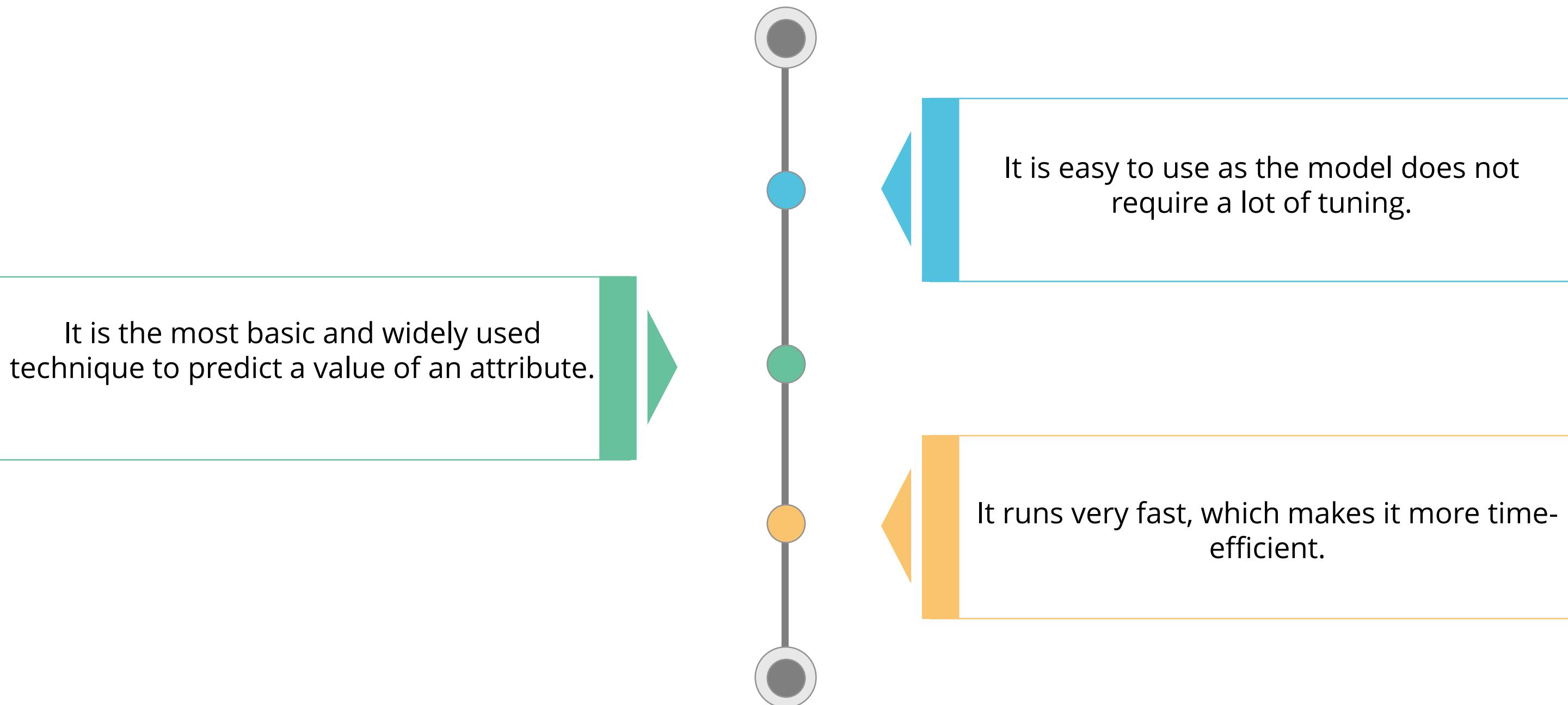


The correct answer is **a**.

**Explanation:** The estimator instance or object is a model.

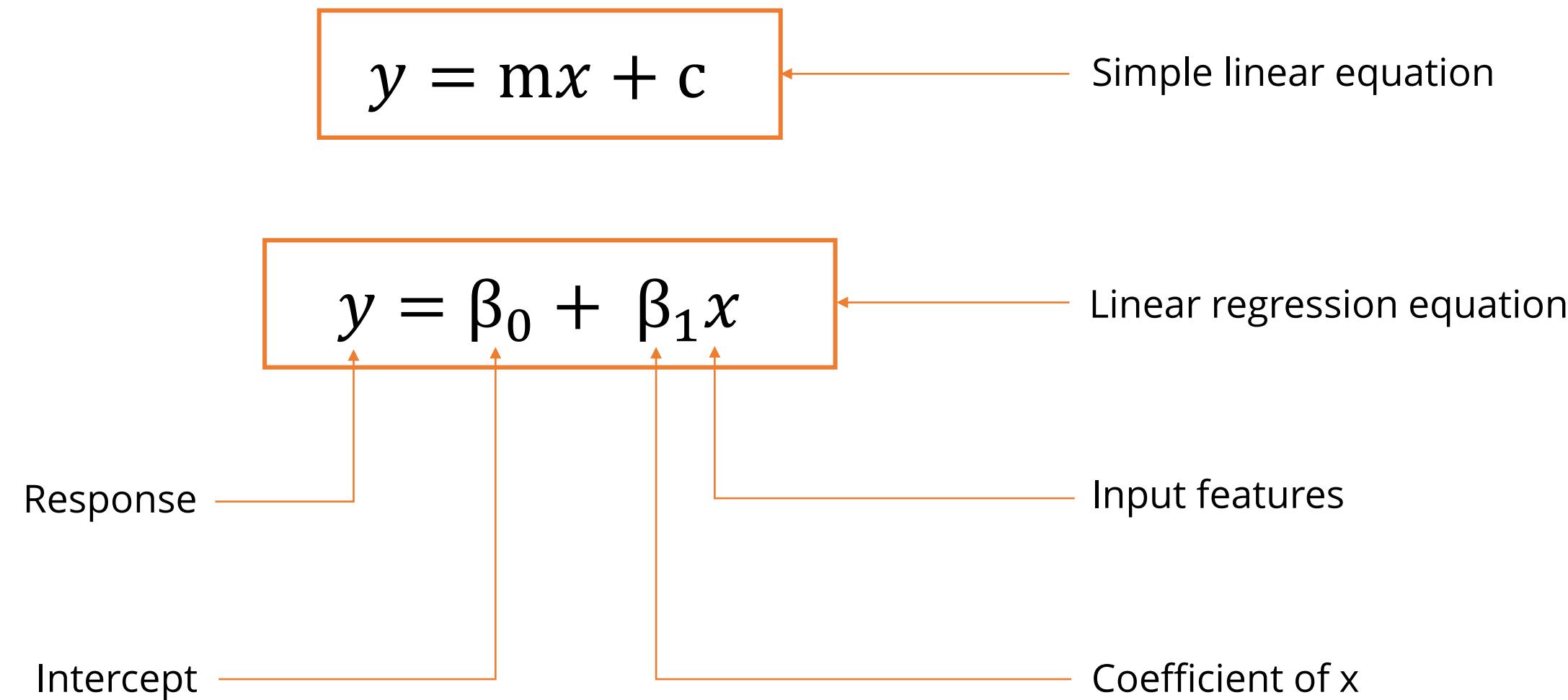
# Supervised Learning Models: Linear Regression

Linear regression is a supervised learning model used to analyze continuous data.



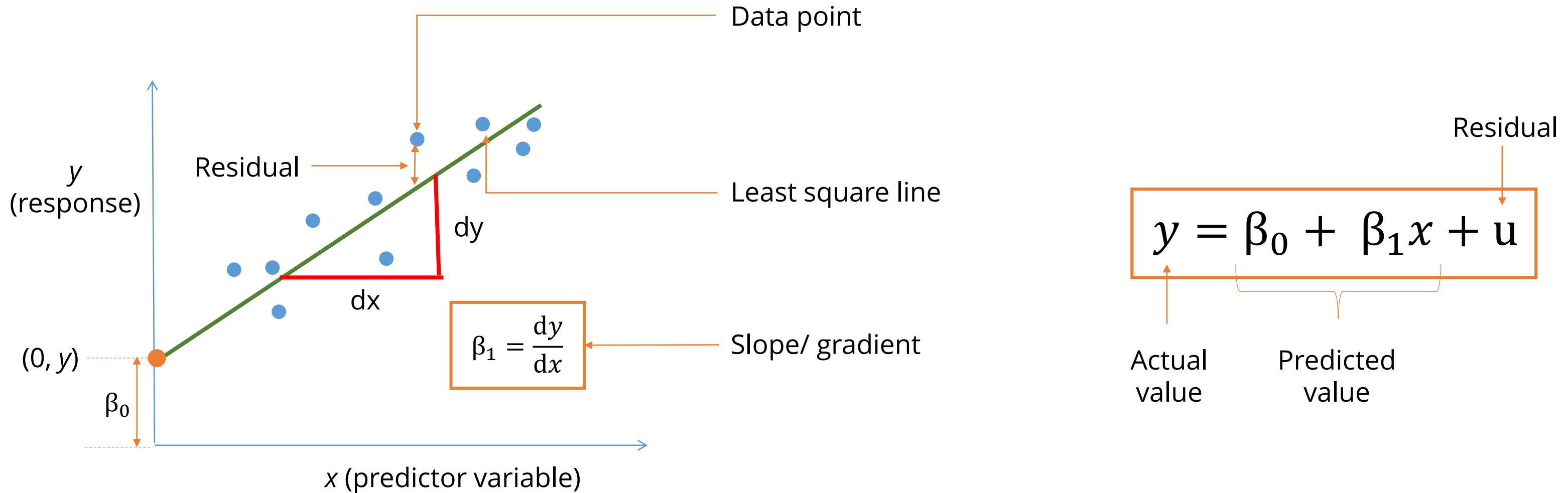
# Supervised Learning Models: Linear Regression (contd.)

The linear regression equation is based on the formula for a simple linear equation.



# Supervised Learning Models: Linear Regression (contd.)

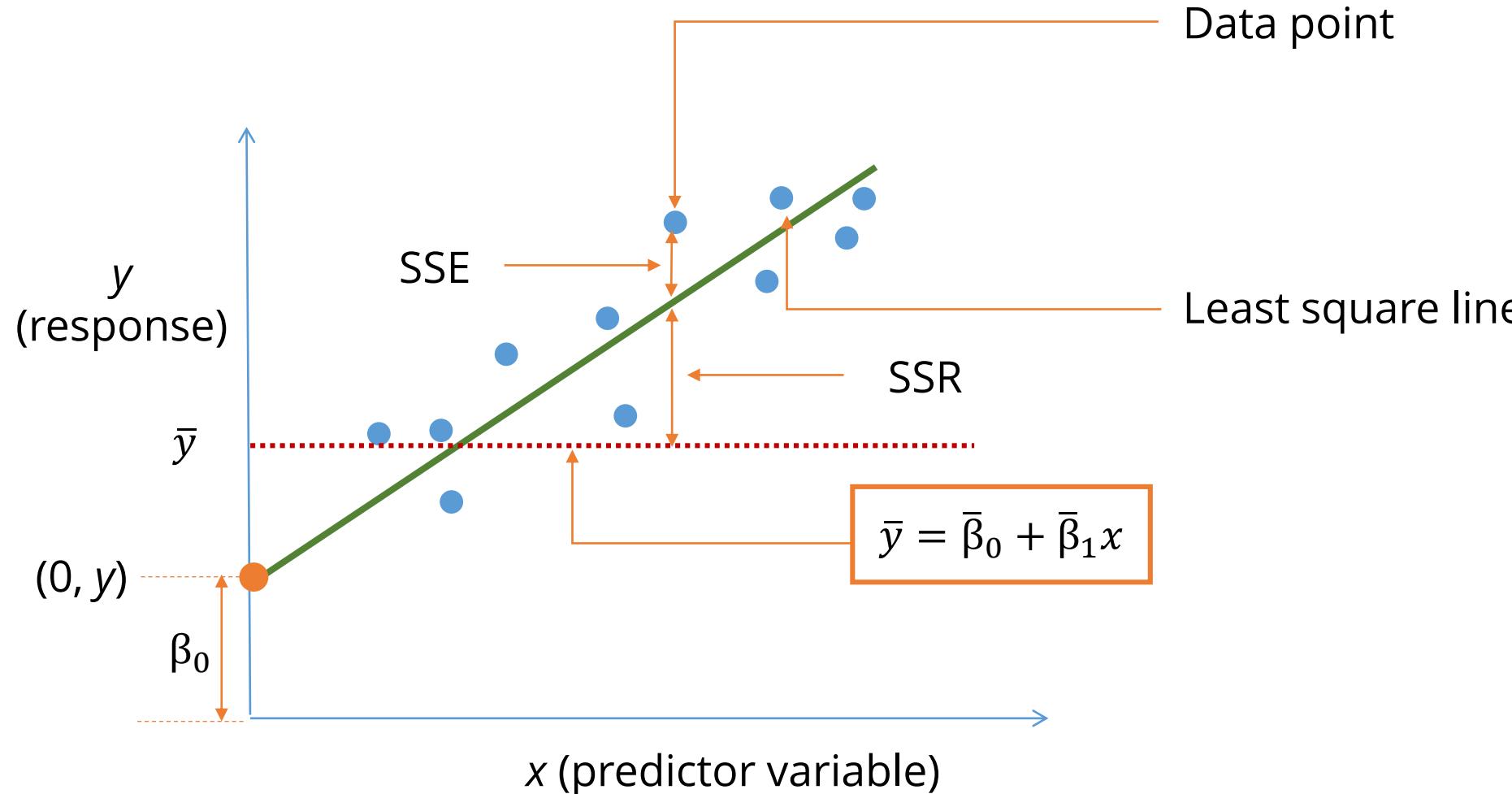
Linear regression is the most basic technique to predict a value of an attribute.



The attributes are usually fitted using the “least square” approach.

# Supervised Learning Models: Linear Regression (contd.)

Smaller the value of SSR or SSE, the more accurate the prediction will be, which would make the model **the best fit**.



$$y = \beta_0 + \beta_1 x + u$$

$$\text{SSR} = \sum (\hat{y}_i - \bar{y})^2$$

Regression of sum of squares

$$\text{SSE} = \sum (y_i - \hat{y}_i)^2$$

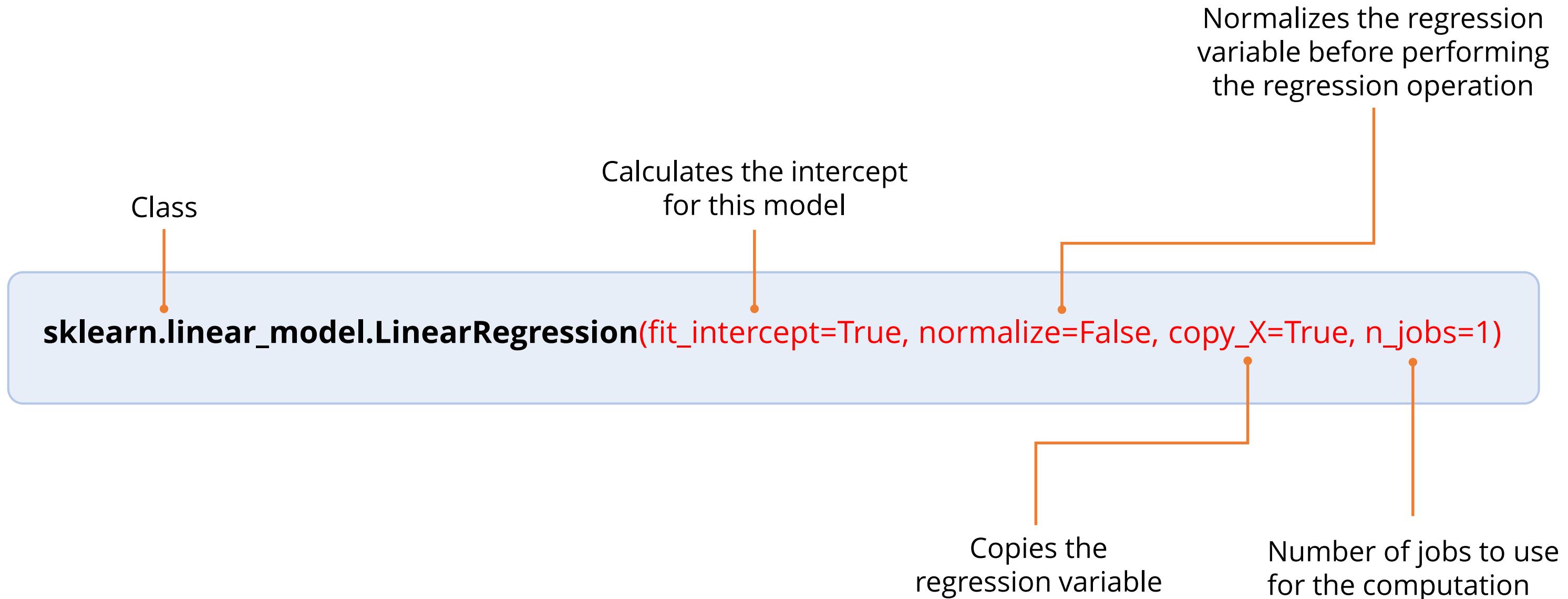
Error of sum of squares



The attributes are usually fitted using the “least square” approach.

# Supervised Learning Models: Linear Regression (contd.)

Let us see how linear regression works in Scikit-Learn.





## Demo 01—Loading a Dataset

Demonstrate how to load a built-in scikit-learn dataset

DATA  
SCIENCE



## Demo 02—Linear Regression Model

Demonstrate how to create and train a linear regression model

DATA  
SCIENCE

# Supervised Learning Models: Logistic Regression

Logistic regression is a generalization of the linear regression model used for classification problems.

$$\pi = \Pr(y = 1|x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Probability of  $y = 1$ , given  $x$

Change in the log-odds  
for a unit change in  $x$



The purpose of K-NN is to predict the class for each observation.

## Supervised Learning Models: Logistic Regression (contd.)

Logistic regression is a generalization of the linear regression model used for classification problems.

$$\text{Odds} = \frac{\pi}{1 - \pi}$$

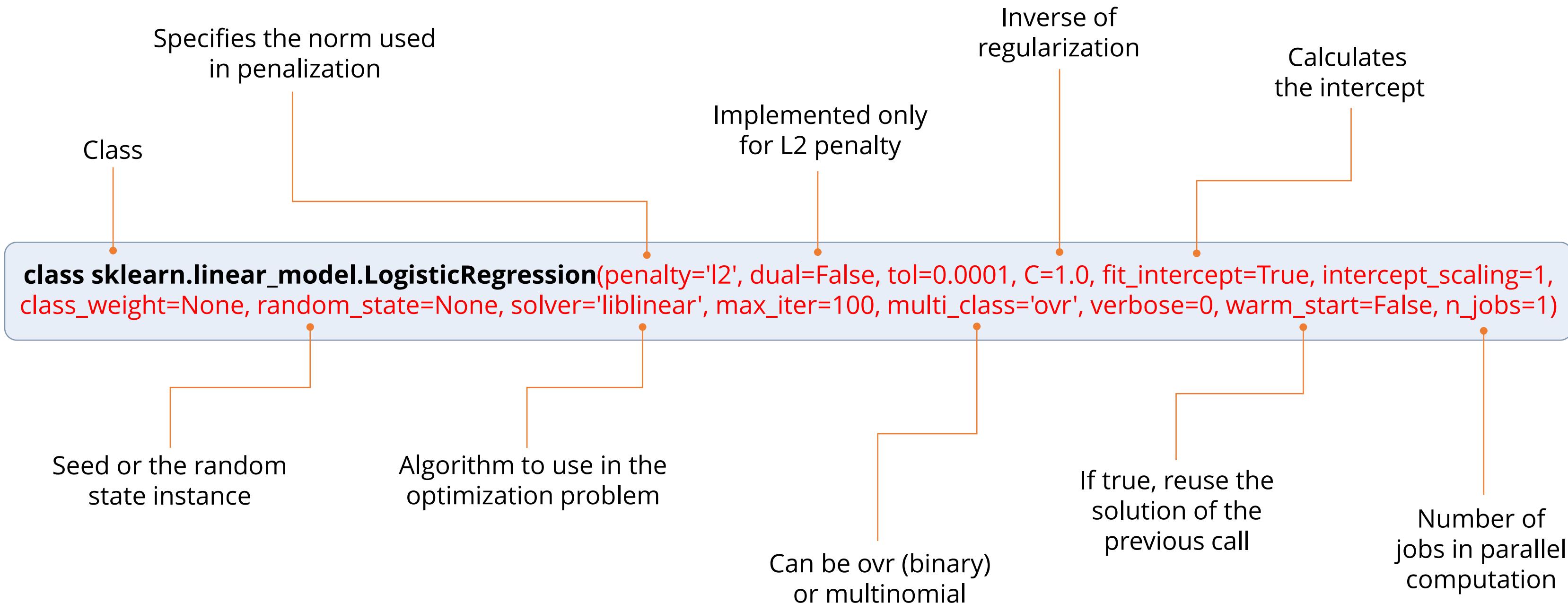
Probability 

$$\log\left(\frac{\pi}{1 - \pi}\right) = \log(e^{\beta_0 + \beta_1 x}) = \beta_0 + \beta_1 x$$

Logarithm of odds  Linear regression 

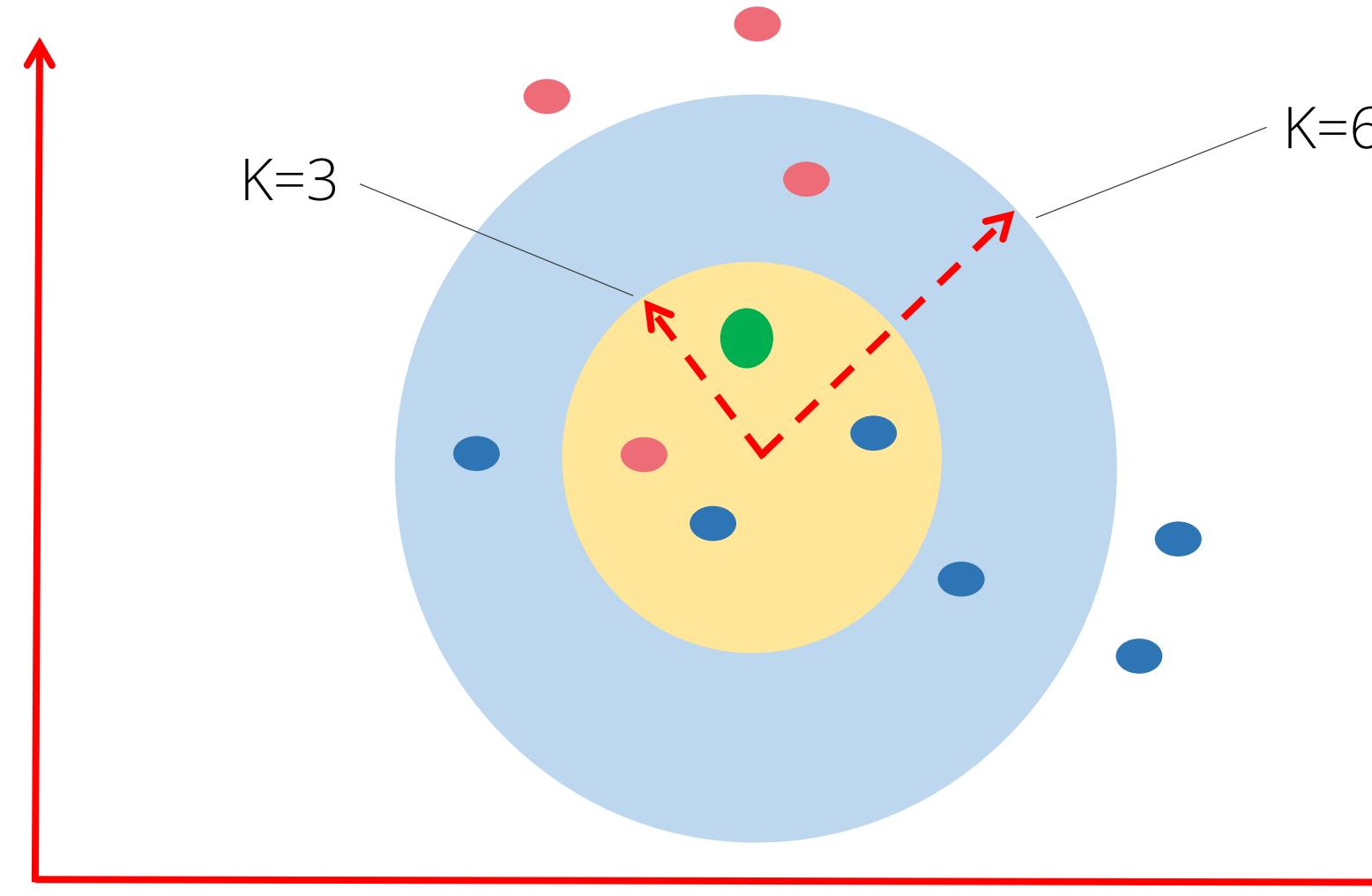
# Supervised Learning Models: Logistic Regression (contd.)

Logistic regression is a generalization of the linear regression model used for classification problems.



# Supervised Learning Models: K Nearest Neighbors (K-NN)

K-nearest neighbors, or K-NN, is one of the simplest machine learning algorithms used for both classification and regression problem types.



If you are using this method for binary classification, choose an odd number for k to avoid the case of a "tied" distance between two classes.



## Demo 03—K-NN and Logistic Regression Models

Demonstrate the use of K-NN and logistic regression models

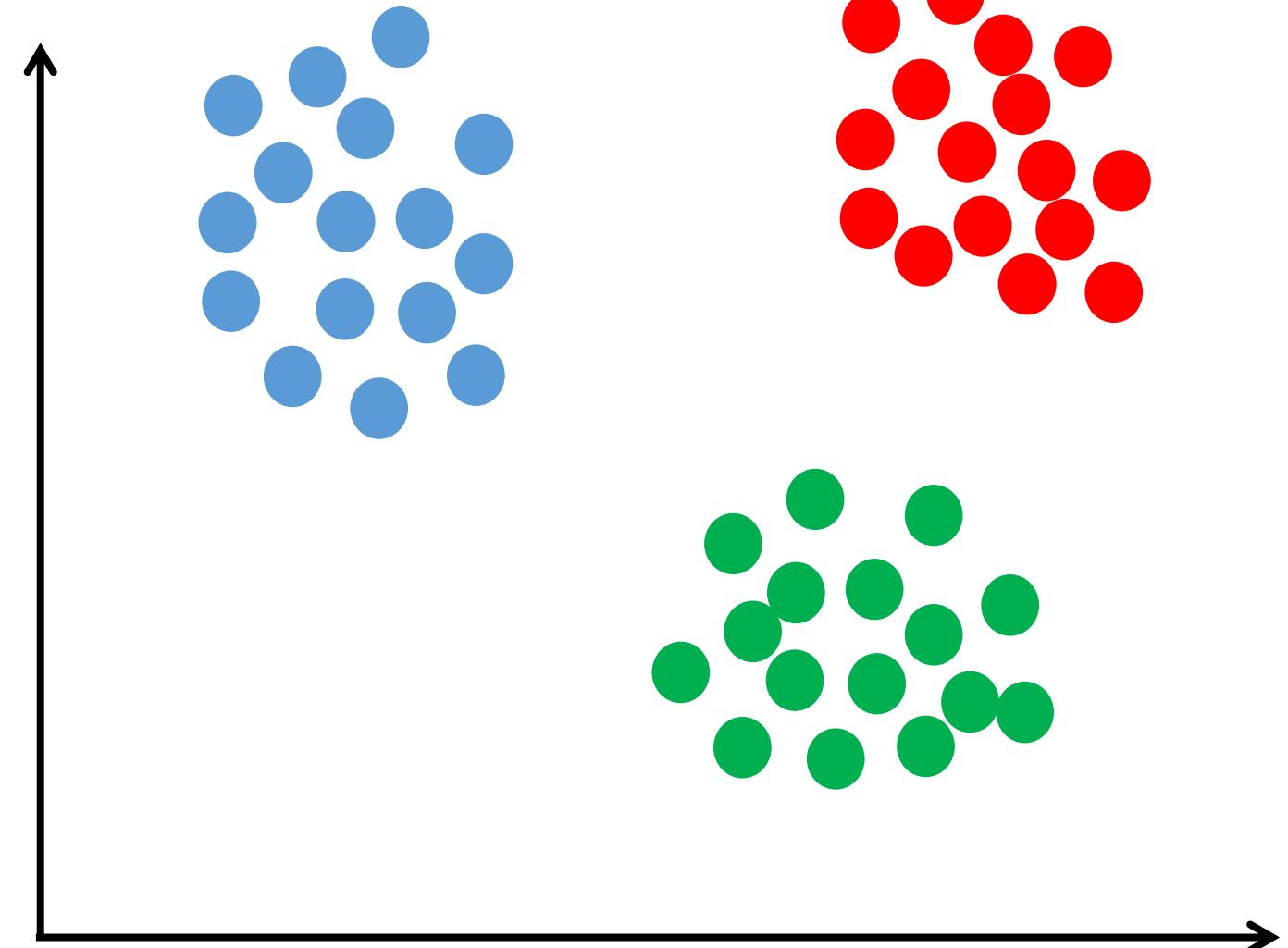
DATA  
SCIENCE

# Unsupervised Learning Models: Clustering

A cluster is a group of similar data points.

It is used:

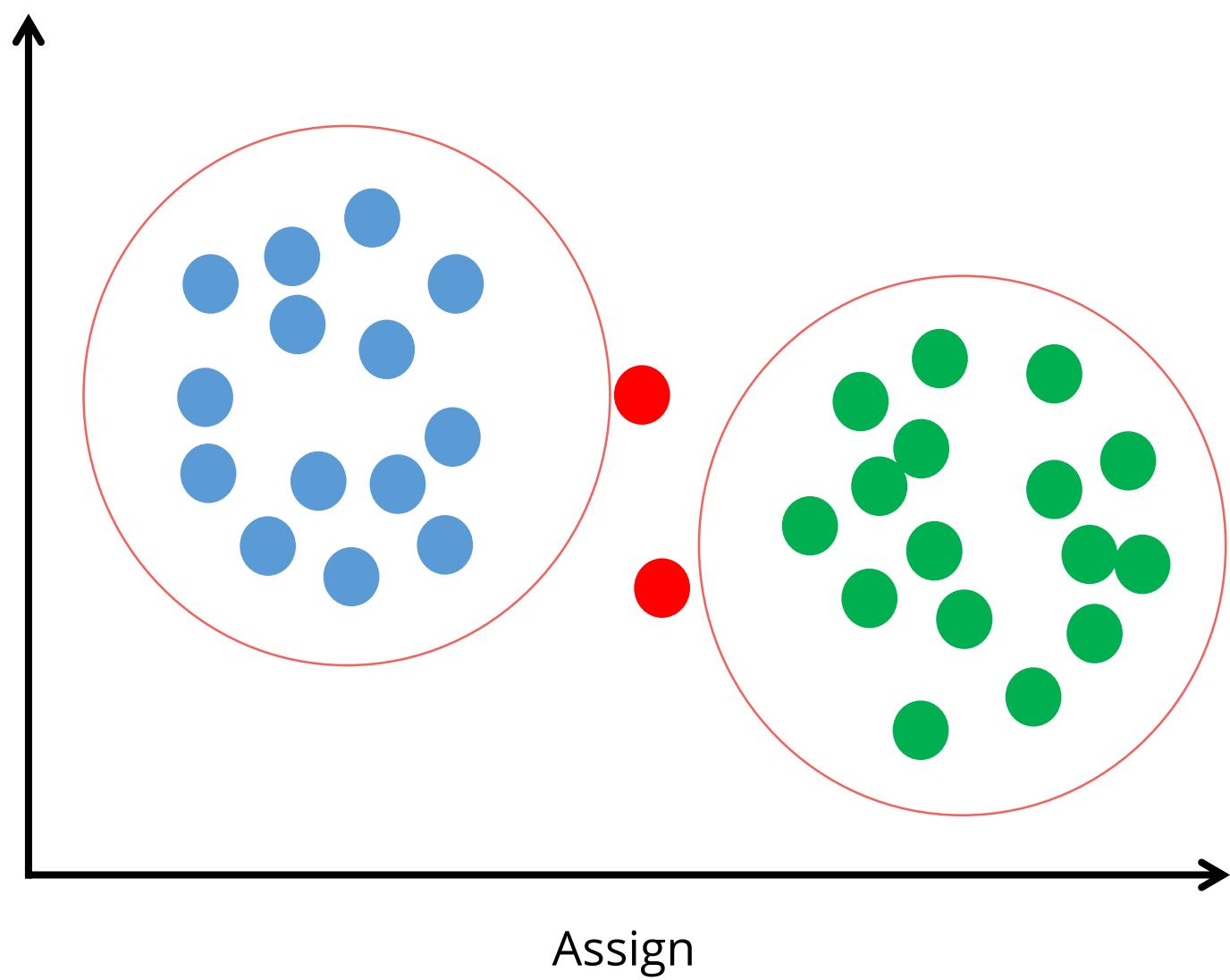
- To extract the structure of the data
- To identify groups in the data



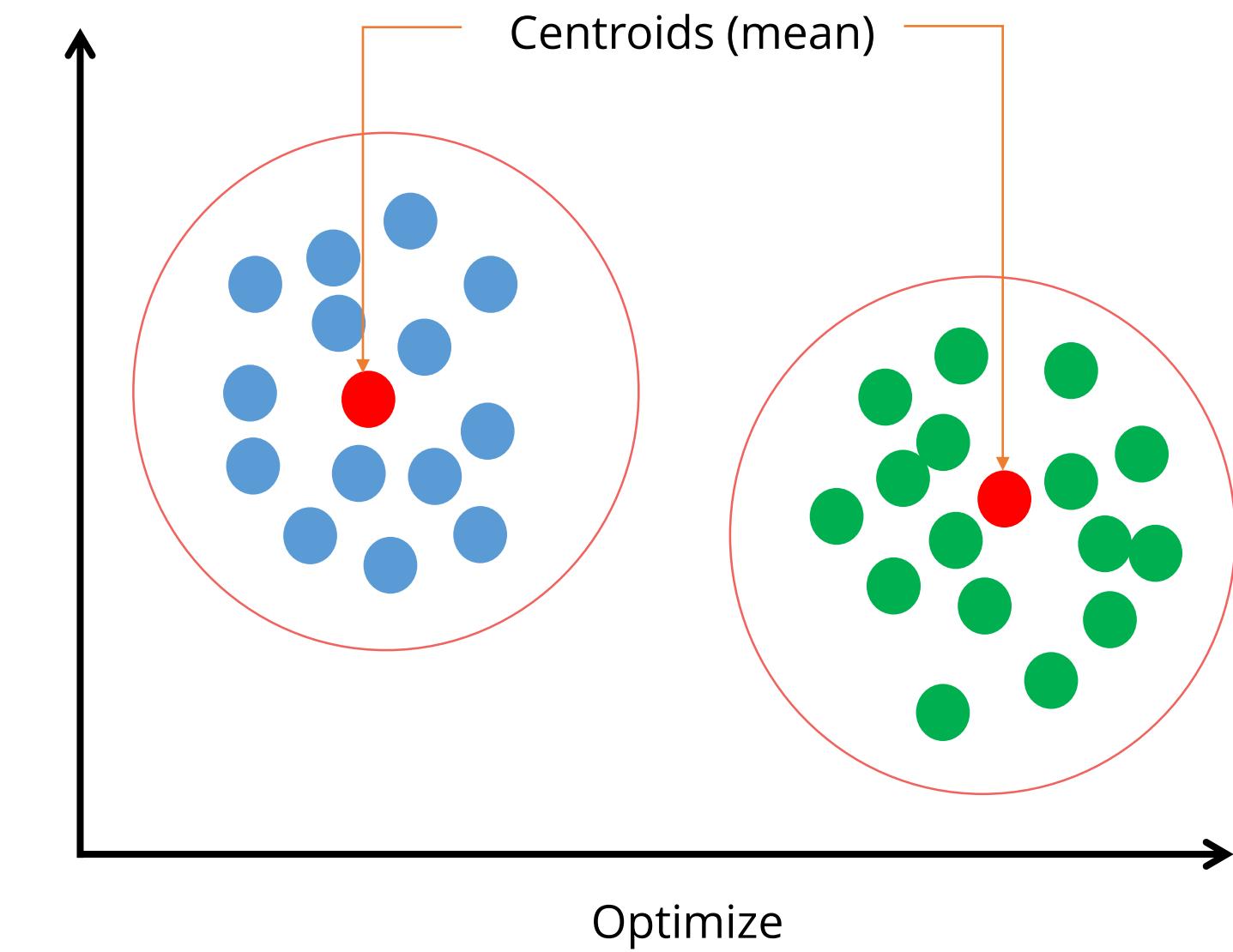
Greater similarity between data points results in better clustering.

# Unsupervised Learning Models: K-means Clustering

K-means finds the best centroids by alternatively assigning random centroids to a dataset and selecting mean data points from the resulting clusters to form new centroids. It continues this process iteratively until the model is optimized.



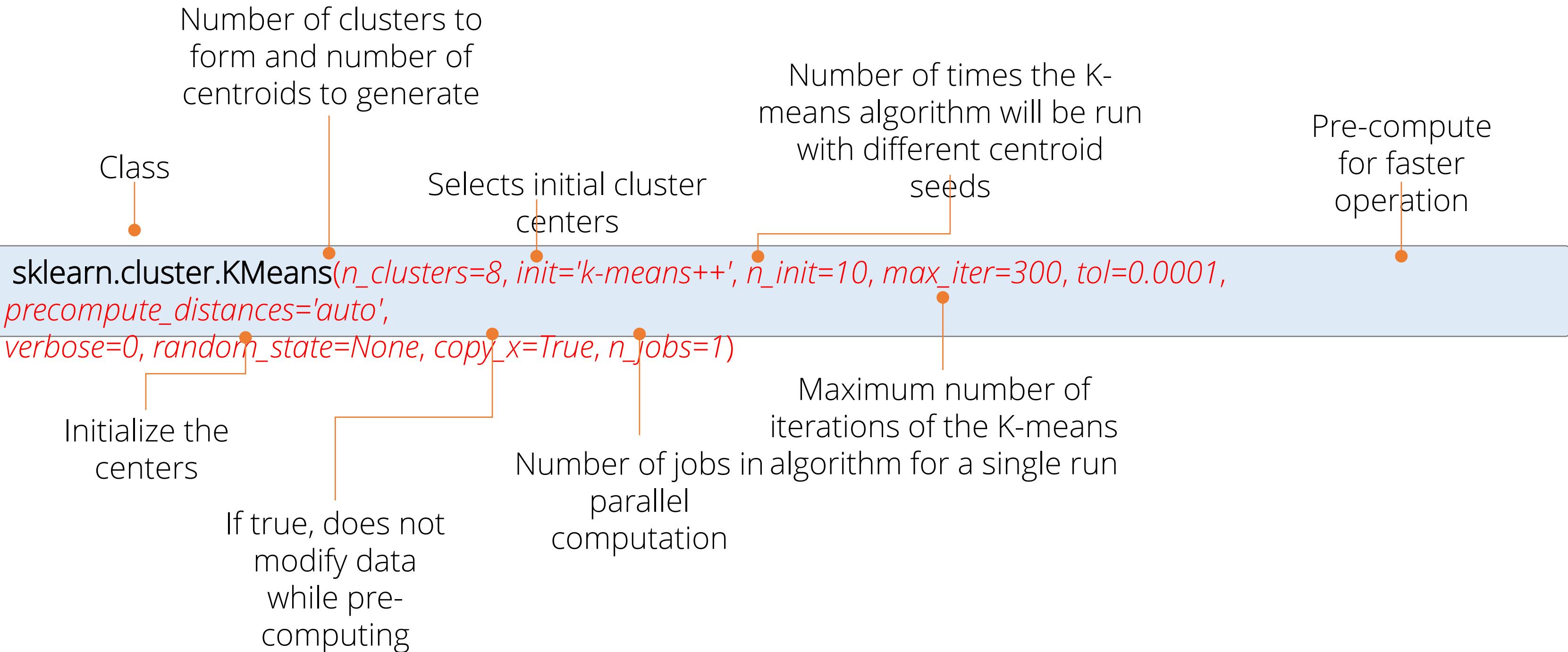
Find the number of clusters and assign mean



Iterate and optimize the mean for each cluster for its respective data points

# Unsupervised Learning Models: K-means Clustering (contd.)

Let us see how the k-means algorithm works in Scikit-Learn.





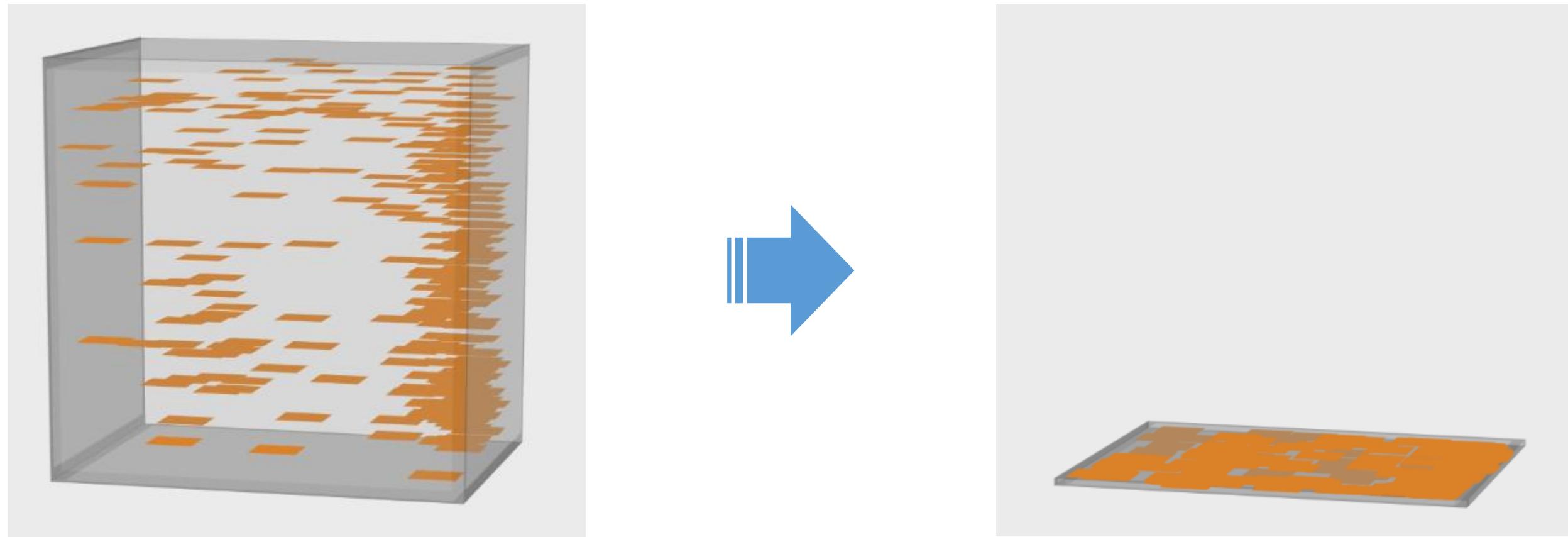
## Demo 04—K-means Clustering

Demonstrate how to use k-means clustering to classify data points

DATA  
SCIENCE

# Unsupervised Learning Models: Dimensionality Reduction

It reduces a high-dimensional dataset into a dataset with fewer dimensions. This makes it easier and faster for the algorithm to analyze the data.



# Unsupervised Learning Models: Dimensionality Reduction (contd.)

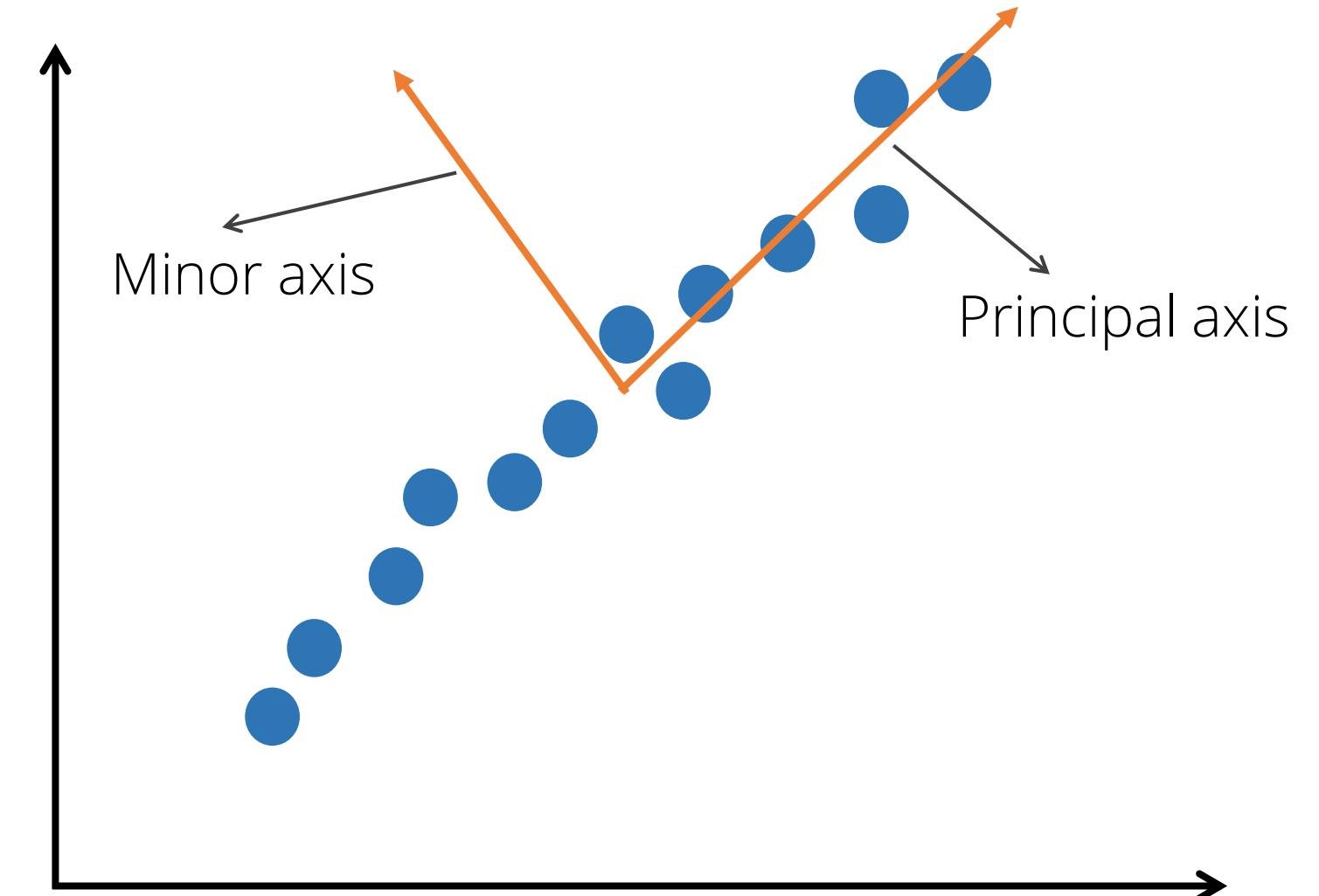
These are some techniques used for dimensionality reduction:

State	Variable	Y1340	Y1341	Y1342	Y1343	Y1344	Y1345	Y1346	Y1347	Y1348	Y1349	Y1350	Y1351	Y1352	Y1353	Y1354	Y1355	Y1356	Y1357	Y1358	Y1359			
Alabama	GDP	0.442683	0.315485	0.109414	0.411838	0.87309	0.531543	0.348278	0.739349	0.705022	0.690683	0.721098	0.853162	0.573774	0.13838	0.037726	0.916111	0.88906	0.653864	0.628792	0.291581			
Alabama	Unemp	0.652003	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alabama	House Price	0.811003	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alabama	Police Budget	0.578769	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alabama	School Budget	0.578769	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alaska	GDP	0.578769	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alaska	Police Budget	0.745779	0.159603	0.150518	0.310792	0.280225	0.092755	0.020034	0.580123	0.574826	0.181356	0.636117	0.435531	0.477461	0.000432	0.135442	0.050528	0.605278	0.745779	0.159603	0.150518			
Alaska	Unemp	0.647951	0.232988	0.039415	0.263495	0.1697	0.482901	0.127472	0.28212	0.182859	0.318863	0.351703	0.616952	0.018594	0.958324	0.830248	0.745779	0.159603	0.150518	0.605278	0.745779	0.159603	0.150518	
Alaska	House Price	0.340723	0.457659	0.688661	0.359685	0.023767	0.531594	0.35789	0.016201	0.74771	0.116944	0.269702	0.378761	0.838736	0.123653	0.610437	0.63698	0.097339	0.560895	0.666404	0.73167			
Alaska	Prisons	0.038474	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alaska	School Budget	0.048050	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581			
Alaska	Police Budget	0.461253	0.327954	0.377816	0.795453	0.411574	0.655371	0.377504	0.75355	0.380534	0.821111	0.76152	0.11729	0.28015	0.655371	0.677465	0.043703	0.515778	0.366694	0.461253	0.327954	0.377816	0.795453	
Arkansas	GDP	0.735344	0.405681	0.252979	0.160238	0.080534	0.900961	0.627304	0.616188	0.089243	0.089243	0.136397	0.58918	0.34841	0.633931	0.309744	0.399549	0.467553	0.735344	0.405681	0.252979	0.160238		
Arkansas	Unemp	0.592559	0.467733	0.360323	0.411224	0.138199	0.576959	0.113724	0.513229	0.18529	0.596739	0.316518	0.438938	0.415867	0.374687	0.494419	0.133827	0.656382	0.592559	0.467733	0.360323	0.411224	0.138199	
Arkansas	House Price	0.605315	0.232988	0.488503	0.166813	0.356668	0.748936	0.771634	0.083595	0.25212	0.717794	0.209384	0.519883	0.584801	0.125058	0.582149	0.505998	0.380323	0.748936	0.771634	0.083595	0.25212	0.717794	0.209384
Arkansas	Prisons	0.826649	0.613386	0.100428	0.197751	0.890656	0.884806	0.587972	0.049476	0.558629	0.424274	0.589619	0.479477	0.190784	0.347426	0.240542	0.750414	0.272051	0.592559	0.467733	0.360323	0.411224	0.138199	
Arkansas	School Budget	0.038474	0.359898	0.021196	0.377979	0.083906	0.270768	0.725234	0.805016	0.052549	0.250004	0.842095	0.624484	0.897205	0.525048	0.840295	0.270768	0.725234	0.805016	0.052549	0.250004	0.842095	0.624484	
Arkansas	Police Budget	0.038474	0.359898	0.021196	0.377979	0.083906	0.270768	0.725234	0.805016	0.052549	0.250004	0.842095	0.624484	0.897205	0.525048	0.840295	0.270768	0.725234	0.805016	0.052549	0.250004	0.842095	0.624484	
California	GDP	0.449602	0.031370	0.819983	0.560113	0.01183	0.88491	0.522814	0.396244	0.469756	0.859994	0.884487	0.416412	0.412331	0.649514	0.277379	0.147661	0.39119	0.031370	0.449602	0.031370	0.819983	0.560113	
California	Unemp	0.744424	0.636742	0.171813	0.929798	0.478153	0.239275	0.627585	0.126518	0.350107	0.549127	0.102129	0.281344	0.031583	0.478645	0.639713	0.574242	0.880597	0.244628	0.059407	0.277705	0.291581		
California	House Price	0.388862	0.232988	0.898481	0.431472	0.083836	0.595088	0.147377	0.482647	0.598483	0.027076	0.727302	0.829378	0.123894	0.480292	0.640951	0.083107	0.830501	0.388862	0.232988	0.898481	0.431472	0.083836	
California	School Budget	0.999887	0.274743	0.020116	0.128342	0.206524	0.027076	0.727302	0.829378	0.123894	0.480292	0.640951	0.083107	0.830501	0.388862	0.232988	0.898481	0.431472	0.083836	0.27076	0.727302	0.829378	0.123894	
Colorado	GDP	0.340723	0.457659	0.688661	0.359685	0.023767	0.531594	0.35789	0.016201	0.74771	0.116944	0.269702	0.378761	0.838736	0.123653	0.610437	0.63698	0.097339	0.560895	0.666404	0.73167			
Colorado	Unemp	0.250359	0.294742	0.039415	0.263495	0.1697	0.482901	0.880916	0.533979	0.020882	0.822307	0.533979	0.227302	0.822307	0.533979	0.020882	0.822307	0.533979	0.227302	0.822307	0.533979	0.020882	0.822307	
Colorado	House Price	0.592559	0.467733	0.360323	0.411224																			

# Unsupervised Learning Models: Principal Component Analysis (PCA)

It is a linear dimensionality reduction method which uses singular value decomposition of the data and keeps only the most significant singular vectors to project the data to a lower dimensional space.

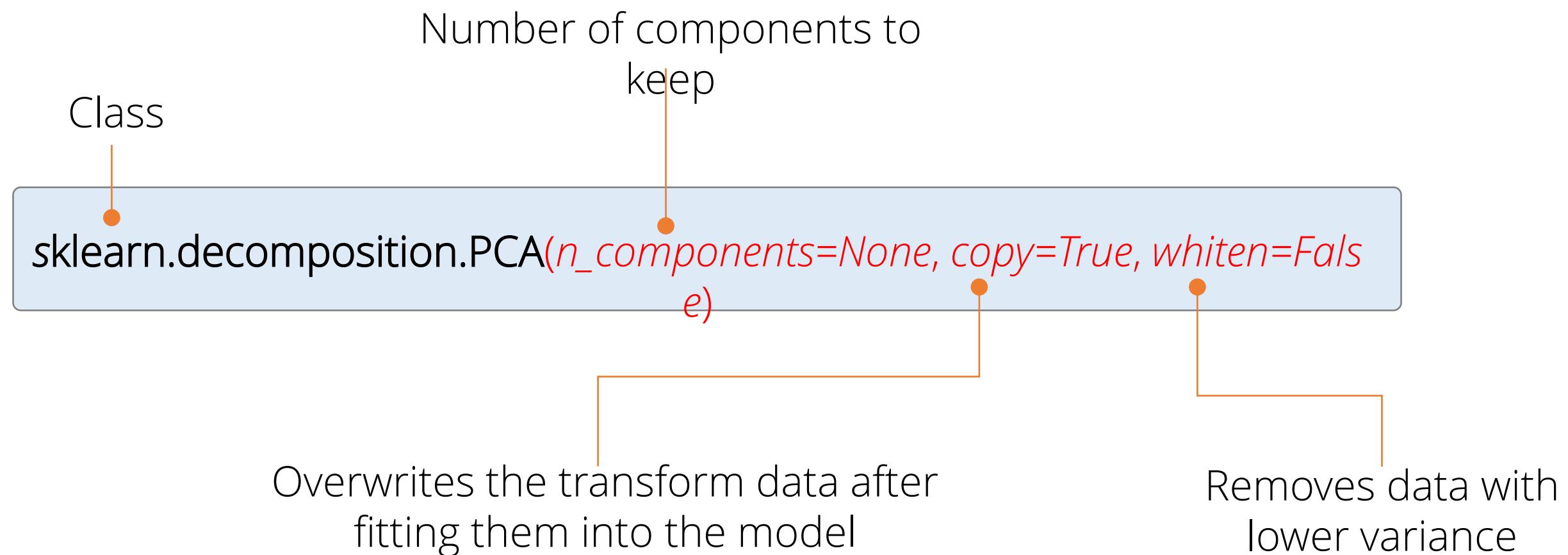
- It is primarily used to compress or reduce the data.
- PCA tries to capture the variance, which helps it pick up interesting features.
- PCA is used to reduce dimensionality in the dataset and to build our feature vector.
- Here, the principal axes in the feature space represents the direction of maximum variance in the data.



This method is used to capture variance.

# Unsupervised Learning Models: Principal Component Analysis (PCA)

Let us look at how the PCA algorithm works in Scikit-Learn.





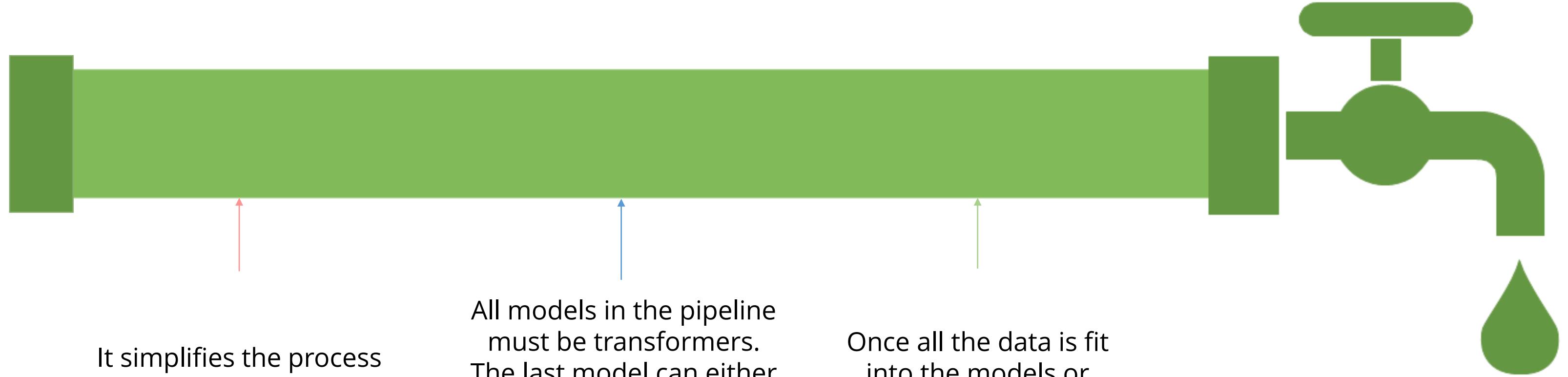
## Demo 05—Principal Component Analysis (PCA)

Demonstrate how to use the PCA model to reduce the dimensions of a dataset

DATA  
SCIENCE

# Pipeline

Pipeline is mainly used to combine multiple models or estimators. Its characteristics are as follows:



It simplifies the process where more than one model is required or used.

All models in the pipeline must be transformers. The last model can either be a transformer or a classifier, regressor, or other such objects.

Once all the data is fit into the models or estimators, the predict method can be called.



Estimators are known as 'model instance'.

## Demo 06—Pipeline

Demonstrate how to build a pipeline



# Model Persistence

Save model for the future use. No need to retrain your model every time when you need them.

It is possible to save a model by using Python's Pickle method.

Scikit-learn has a special replacement for pickle called joblib.

You can use joblib.dump and joblib.load methods.

These are more efficient for Big Data.





## Demo 06—Model Persistence

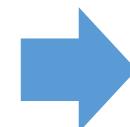
Demonstrate how to persist a model for future use

DATA  
SCIENCE

# Model Evaluation: Metric Functions

You can use the “Metrics” function to evaluate the accuracy of your model’s predictions.

Classification



metrics.**accuracy\_score**  
metrics.**average\_precision\_score**

Clustering



metrics.**adjusted\_rand\_score**

Regression



metrics.**mean\_absolute\_error**  
metrics.**mean\_squared\_error**  
metrics.**median\_absolute\_error**



# Knowledge Check

## What is the best way to train a model?

- a. Use the entire dataset as a training and testing set
- b. Split the known dataset into separate training and testing sets
- c. Ask the source to provide continuous data
- d. Ask the source to provide categorical data



## What is the best way to train a model?

- a. Use the entire dataset as a training and testing set both
- b. Split the known dataset into separate training and testing sets
- c. Ask the source to provide continuous data
- d. Ask the source to provide categorical data



The correct answer is **b**.

**Explanation:** The best way to train a model is to split the known dataset into training and testing sets. The testing set varies from 20% to 40%.



Problem

Instructions

The given dataset contains ad budgets for different media channels and the corresponding ad sales of XYZ firm. Evaluate the dataset to:

- Find the features or media channels used by the firm
- Find the sales figures for each channel
- Create a model to predict the sales outcome
- Split as training and testing datasets for the model
- Calculate the Mean Square Error (MSE)

Problem

Instruction  
s

Instructions on performing the assignment:

- Download the “Advertising Budget and Sales.csv” file from the “Resource” tab. You can load the saved file to the Jupyter notebook that you would be using to complete the assignment.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



Problem

Instructions

The given dataset lists the glucose level readings of several pregnant women taken either during a survey examination or routine medical care. It specifies if the 2 hour post-load plasma glucose was at least 200 mg/dl. Analyze the dataset to:

1. Find the features of the dataset,
2. Find the response label of the dataset,
3. Create a model to predict the diabetes outcome,
4. Use training and testing datasets to train the model, and
5. Check the accuracy of the model.

Problem

Instructions

Instructions on performing the assignment:

- Download the “pima-indians-diabetes.DATA” and “pima-indians-diabetes.NAMES” files from the “Resources” tab. Load the .DATA file to the Jupyter notebook to work on it.
- Open the .NAMES file with a notepad application to view its text. Use this file to view the features of the dataset and add them manually in your code.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



**QUIZ****1**

**Which of the following is true with a greater value of SSR or SSE? *Select all that apply.***

- a. The prediction will be more accurate, making it the best fit model.
- b. The prediction will start becoming less accurate.
- c. The outcome remains unaffected.
- d. The model will not be the best fit for the attributes.



**QUIZ****1**

**Which of the following is true with a greater value of SSR or SSE? Select all that apply.**

- a. The prediction will be more accurate, making it the best fit model.
- b. The prediction will start becoming less accurate.
- c. The outcome remains unaffected.
- d. The model will not be the best fit for the attributes.



The correct answer is **b, d**.

**Explanation:** With higher SSR or SSE, the prediction will be less accurate and the model will not be the best fit for the attributes.

**QUIZ****2**

**Class `sklearn.linear_model.LogisticRegression`, `random_state` \_\_\_\_.**

- a. indicates the seed of the pseudo random number generator used to shuffle data
- b. defines the features state
- c. represents the number of random iterations
- d. specifies a random constant to be added to the decision function



**QUIZ**  
**2**

**Class `sklearn.linear_model.LogisticRegression`, random\_state \_\_\_\_.**

- a. indicates the seed of the pseudo random number generator used to shuffle data
- b. defines the features state
- c. represents the number of random iterations
- d. specifies a random constant to be added to the decision function



The correct answer is **a.**

**Explanation:** The class “`sklearn.linear_model.LogisticRegression`, random\_state” indicates the seed of the pseudo random number generator used to shuffle data.

**QUIZ****3**

**What are the requirements of the K-means algorithm? *Select all that apply.***

- a. Number of clusters should be specified
- b. More than one iteration should meet requisite criteria
- c. Centroids should minimize inertia
- d. Features should be labeled



## QUIZ

3

**What are the requirements of the K-means algorithm? *Select all that apply.***

- a. Number of clusters should be specified
- b. More than one iteration should meet requisite criteria
- c. Centroids should minimize inertia
- d. Features should be labeled



The correct answer is **a, b, c.**

**Explanation:** The K-means algorithm requires that the number of clusters be specified and that centroids that minimize inertia be selected. It requires several iterations to fine tune itself and meet the required criteria to become the best fit model.

**QUIZ**  
**4**

In Class **sklearn.decomposition.PCA**, the **transform(X)** method , where X is multi-dimensional \_\_\_\_.

- a. fits the model with X and applies the dimensionality reduction on X
- b. transforms the data back to its original space
- c. applies the dimensionality reduction on X
- d. computes data co-variance with the generative model



**QUIZ**  
**3**

In Class **sklearn.decomposition.PCA**, the **transform(X)** method , where X is multi-dimensional \_\_\_\_.

- a. fits the model with X and applies the dimensionality reduction on X
- b. transforms the data back to its original space
- c. applies the dimensionality reduction on X
- d. computes data co-variance with the generative model

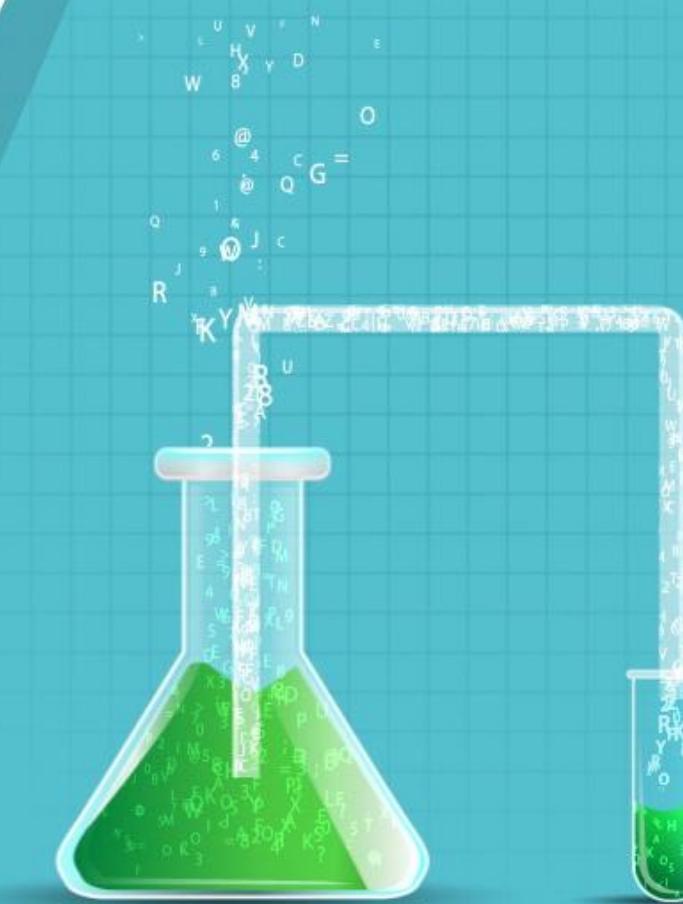


The correct answer is **c.**

**Explanation:** In Class “`sklearn.decomposition.PCA`,” the `transform(X)` method applies the dimensionality reduction on X.

# Key Takeaways

- Scikit-learn has many built-in functions and algorithms which make it easy for Data Scientists to build machine learning models.
- Machine learning can easily and quickly extract information from large sources of data.
- Supervised and unsupervised machine learning models are two of the most widely used learning models.
- Supervised learning models are used to predict the outcome of a dataset.
- Unsupervised learning models are used to find the structure of a dataset.
- For continuous data, you can use regression algorithms if it is a supervised learning model. However, use dimensionality reduction for unsupervised learning.
- For categorical data, use classification algorithms if it is a supervised learning model and clustering if it is an unsupervised learning model.



**This concludes “Machine Learning with Scikit-Learn.”**

The next lesson is “Natural Language Processing (NLP) with Scikit-Learn.”

# Data Science with Python

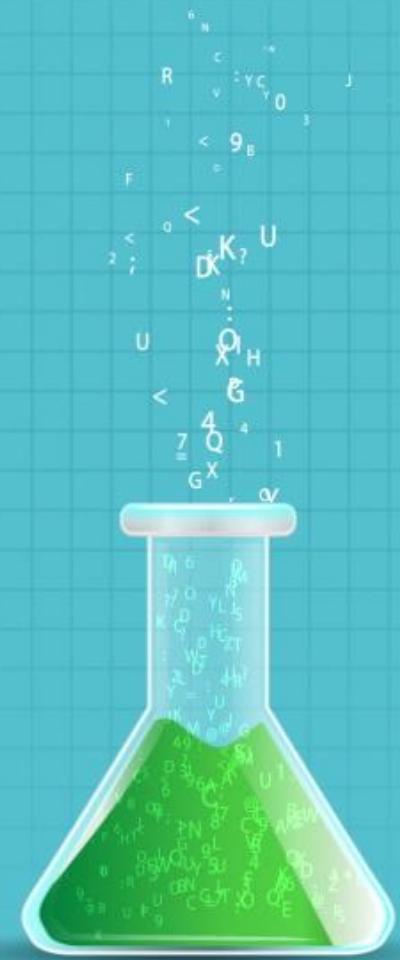
## Lesson 9 — Natural Language Processing (NLP) with SciKit Learn



# What You Will Learn

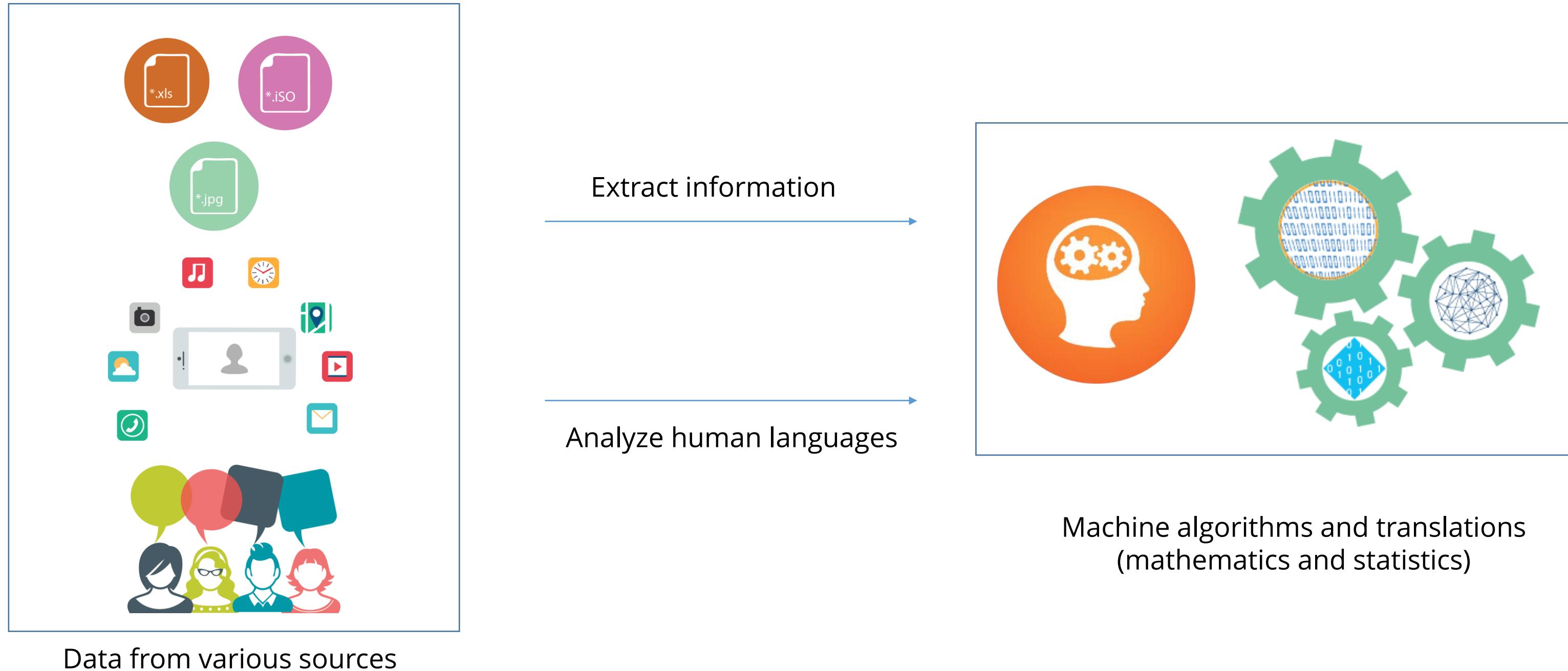
---

- What is Natural Language Processing
- How Natural Language Processing is helpful
- Modules to load content and category
- Applying feature extraction techniques
- Applying approaches of Natural Language Processing



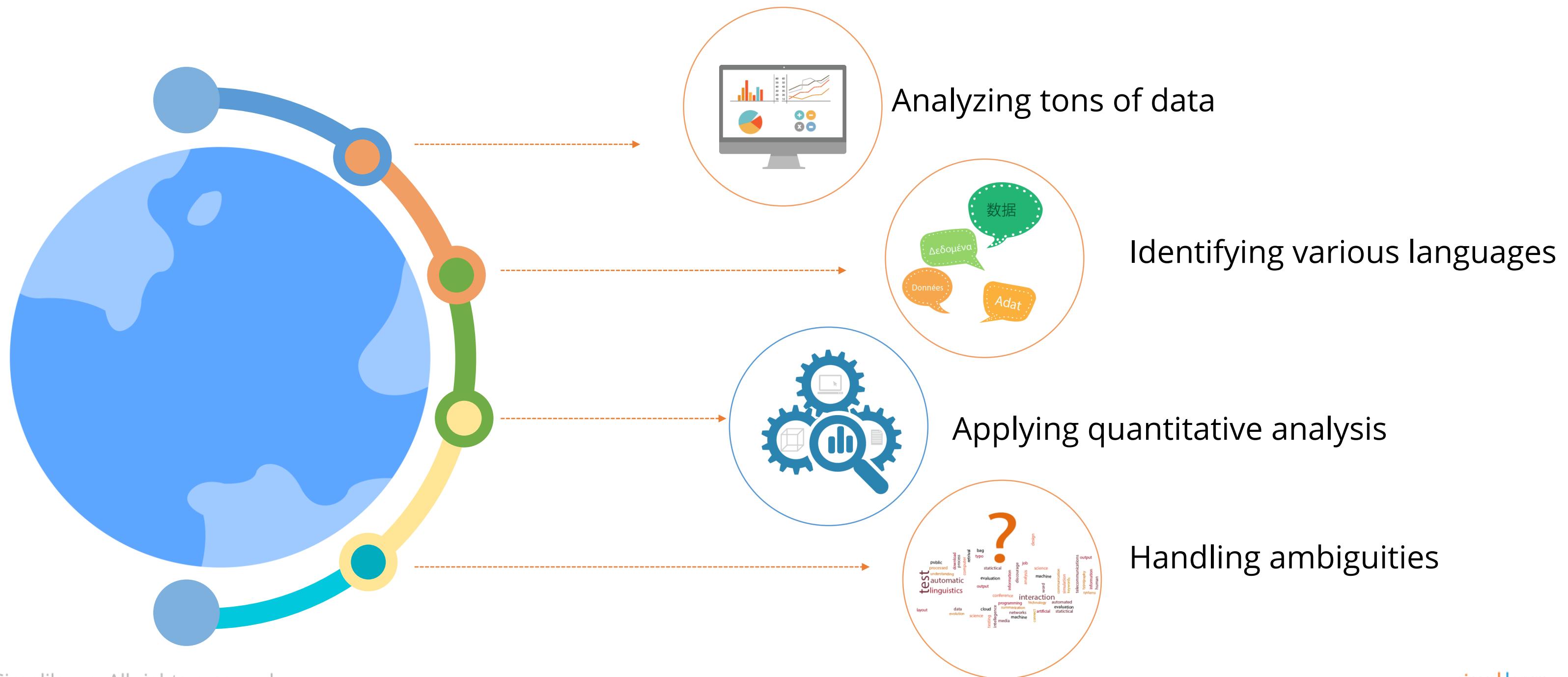
# Natural Language Processing (NLP)

Natural language processing is an automated way to understand and analyze natural human languages and extract information from such data by applying machine algorithms.



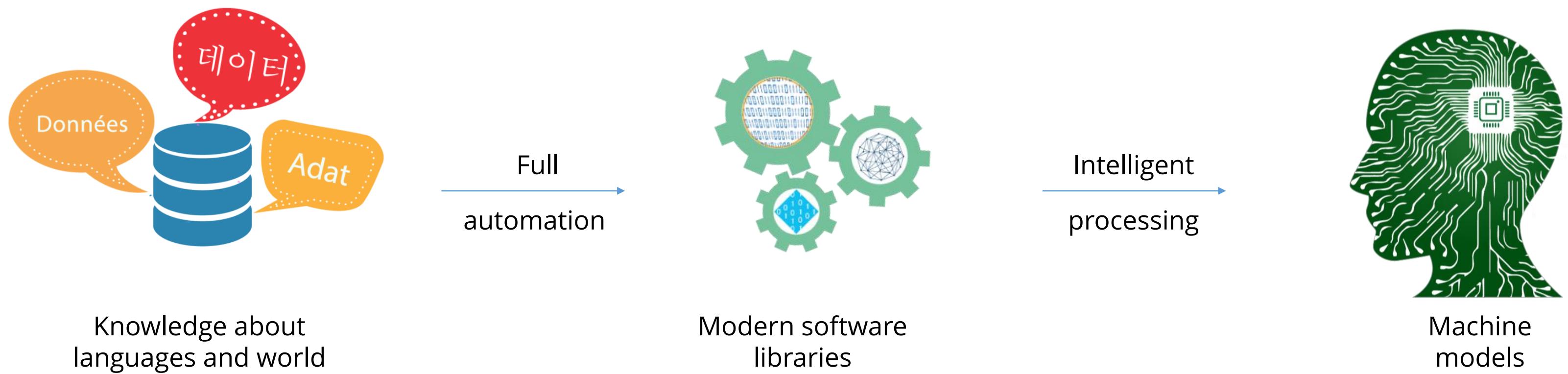
# Why Natural Language Processing

With the advancement in technology and services, the world is now a global village. However, following are a few challenges while analyzing the huge data collection:



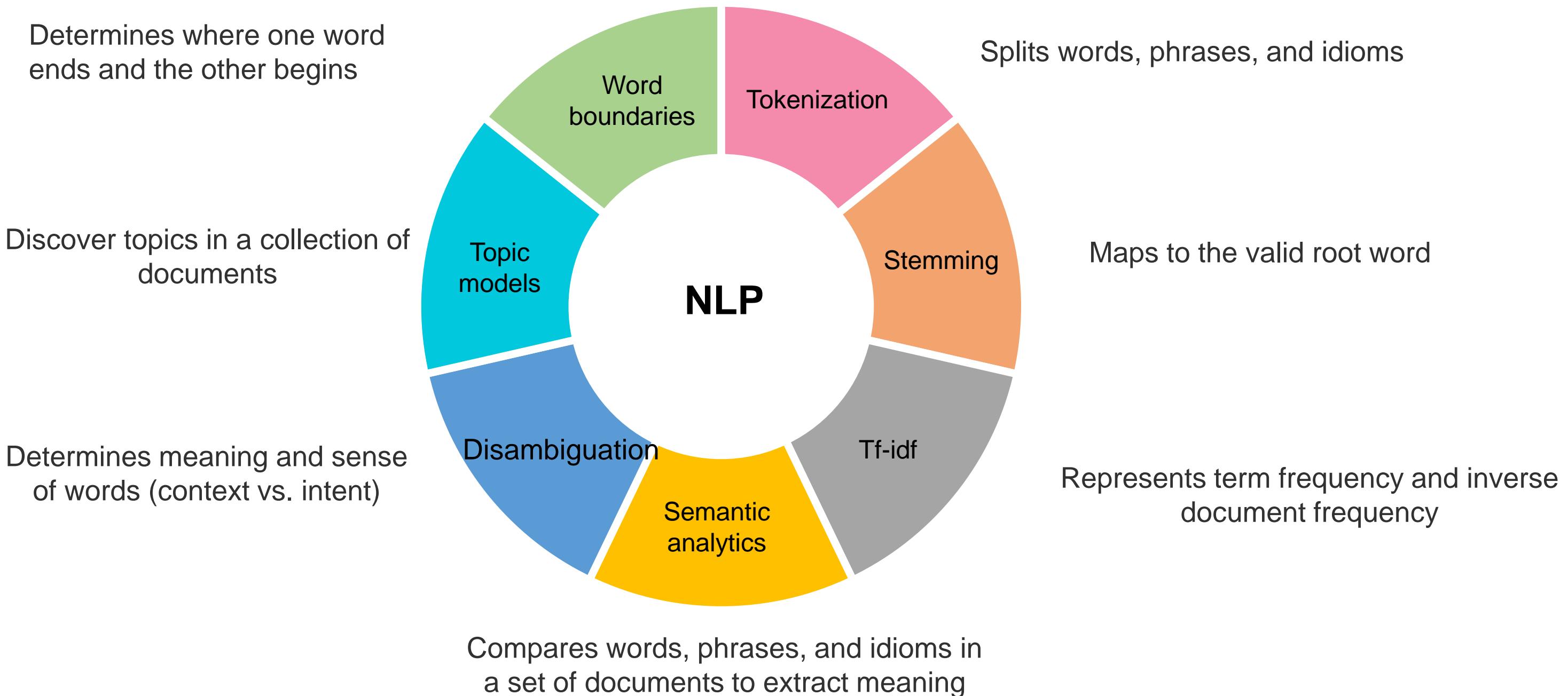
# Why Natural Language Processing (contd.)

In NLP, full automation can be easily achieved by using modern software libraries, modules, and packages.



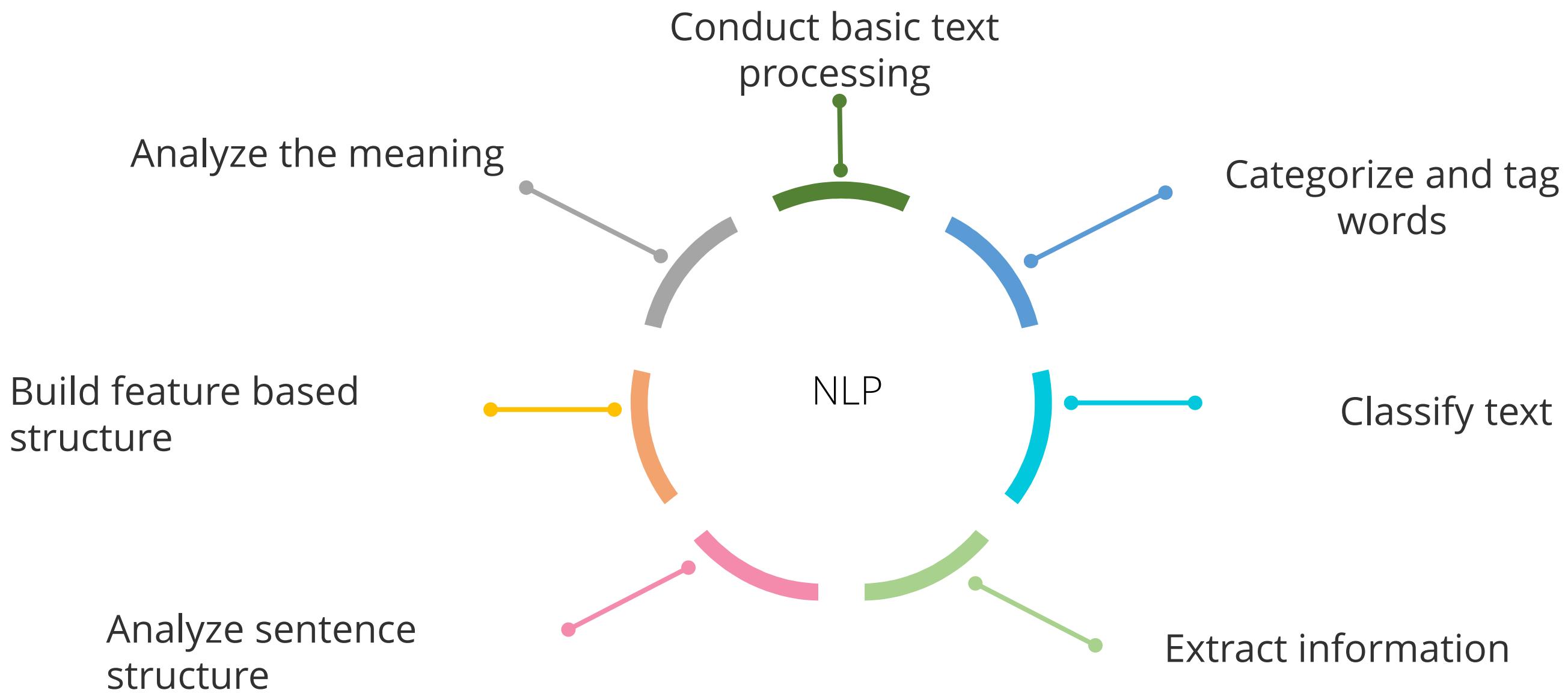
# NLP Terminology

Let us understand the NLP terminologies.



# The NLP Approach for Text Data

Let us look at the Natural Language Processing approaches to analyze text data.





## Demo 01- NLP Environmental Setup

Demonstrate the installation of NLP environment

DATA  
SCIENCE



## Demo 02: Sentence Analysis

Demonstrate the sentence analysis

DATA  
SCIENCE

# The NLP Applications

Let us take a look at the applications that use NLP.

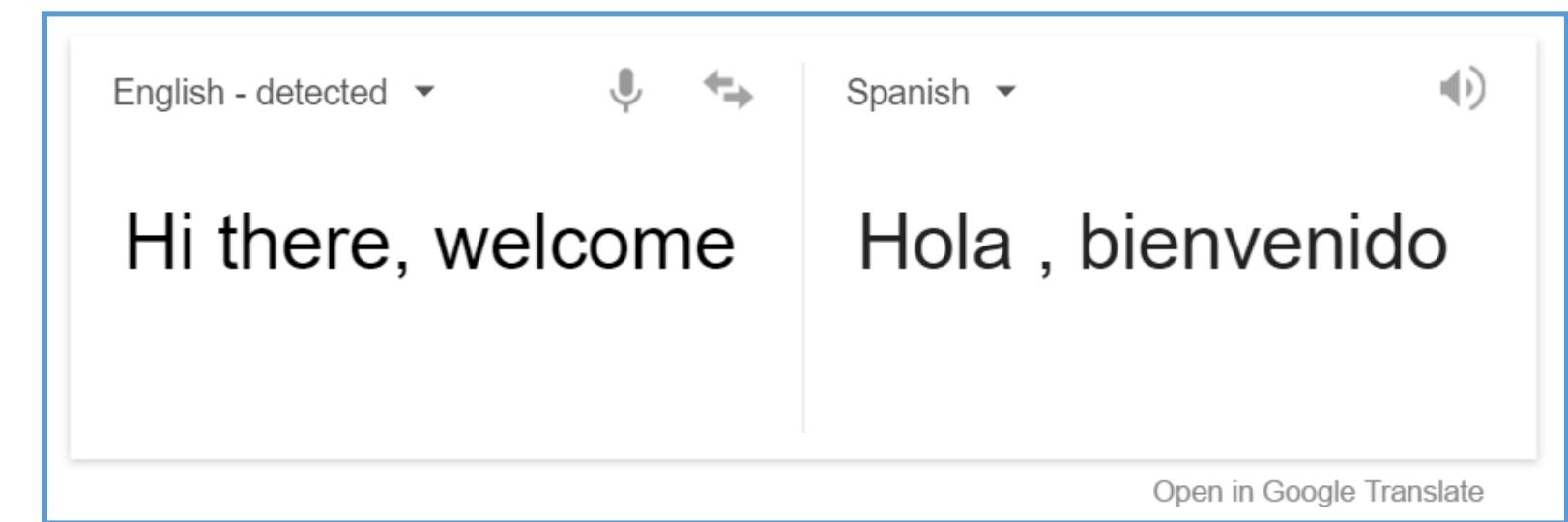
Machine Translation

Speech Recognition

Sentiment Analysis



Machine translation is used to translate one language into another. Google Translate is an example. It uses NLP to translate the input data from one language to another.



# The NLP Applications (contd.)

Let us take a look at the applications that use NLP

Machine Translation

Speech Recognition

Sentiment Analysis

The speech recognition application understands human speech and uses it as input information. It is useful for applications like Siri, Google Now, and Microsoft Cortana.



# The NLP Applications (contd.)

Let us take a look at the applications that use NLP

Machine Translation

Speech Recognition

Sentiment Analysis

Sentiment analysis is achieved by processing tons of data received from different interfaces and sources. For example, NLP uses all social media activities to find out the popular topic of discussion.





KNOWLEDGE  
CHECK

**In NLP, tokenization is a way to**

- a. Find the grammar of the text
- b. Analyze the sentence structure
- c. Find ambiguities
- d. Split text data into words, phrases, and idioms



KNOWLEDGE  
CHECK**In NLP, tokenization is a way to**

- a. Find the grammar of the text
- b. Analyze the sentence structure
- c. Find ambiguities
- d. Split text data into words, phrases, and idioms



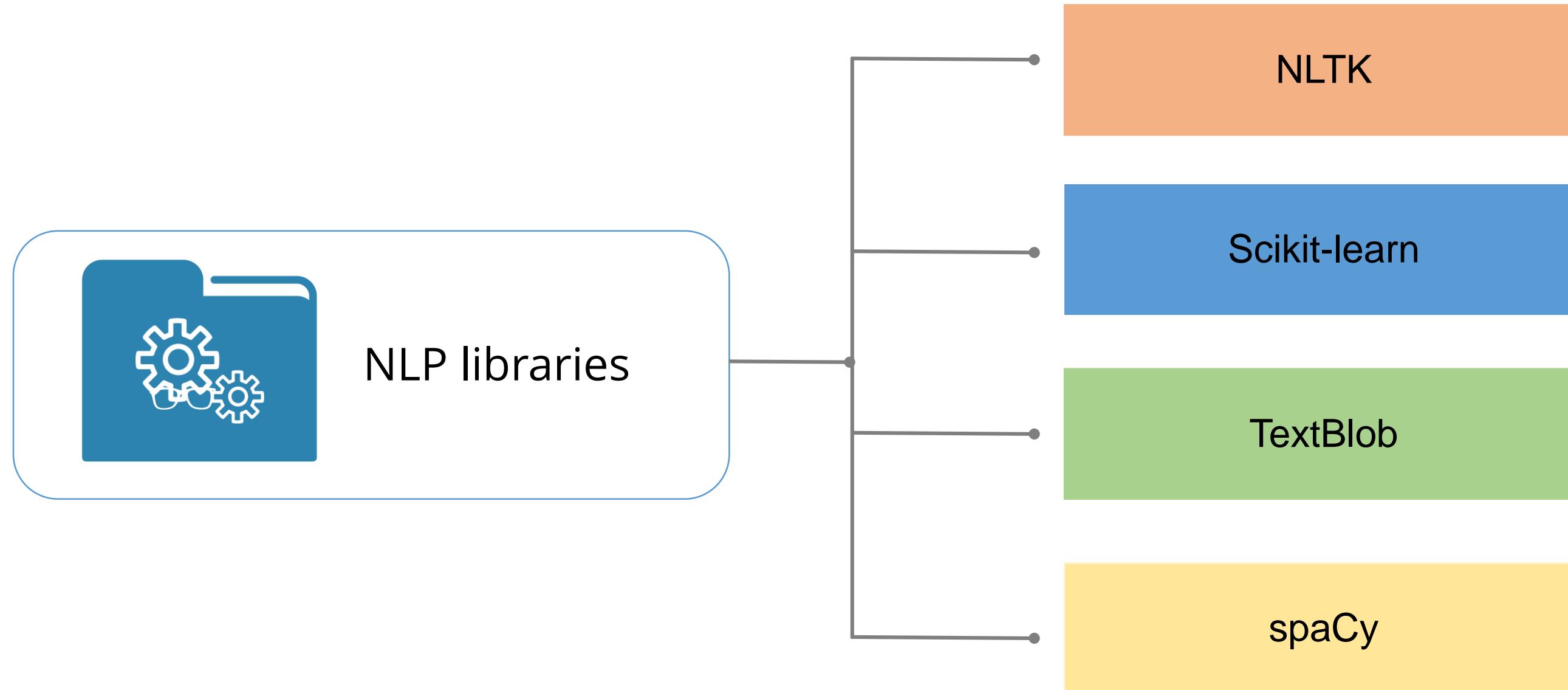
The correct answer is . d

Explanation: Splitting text data into words, phrases, and idioms is known as tokenization and each individual word is known as token.

# Major NLP Libraries

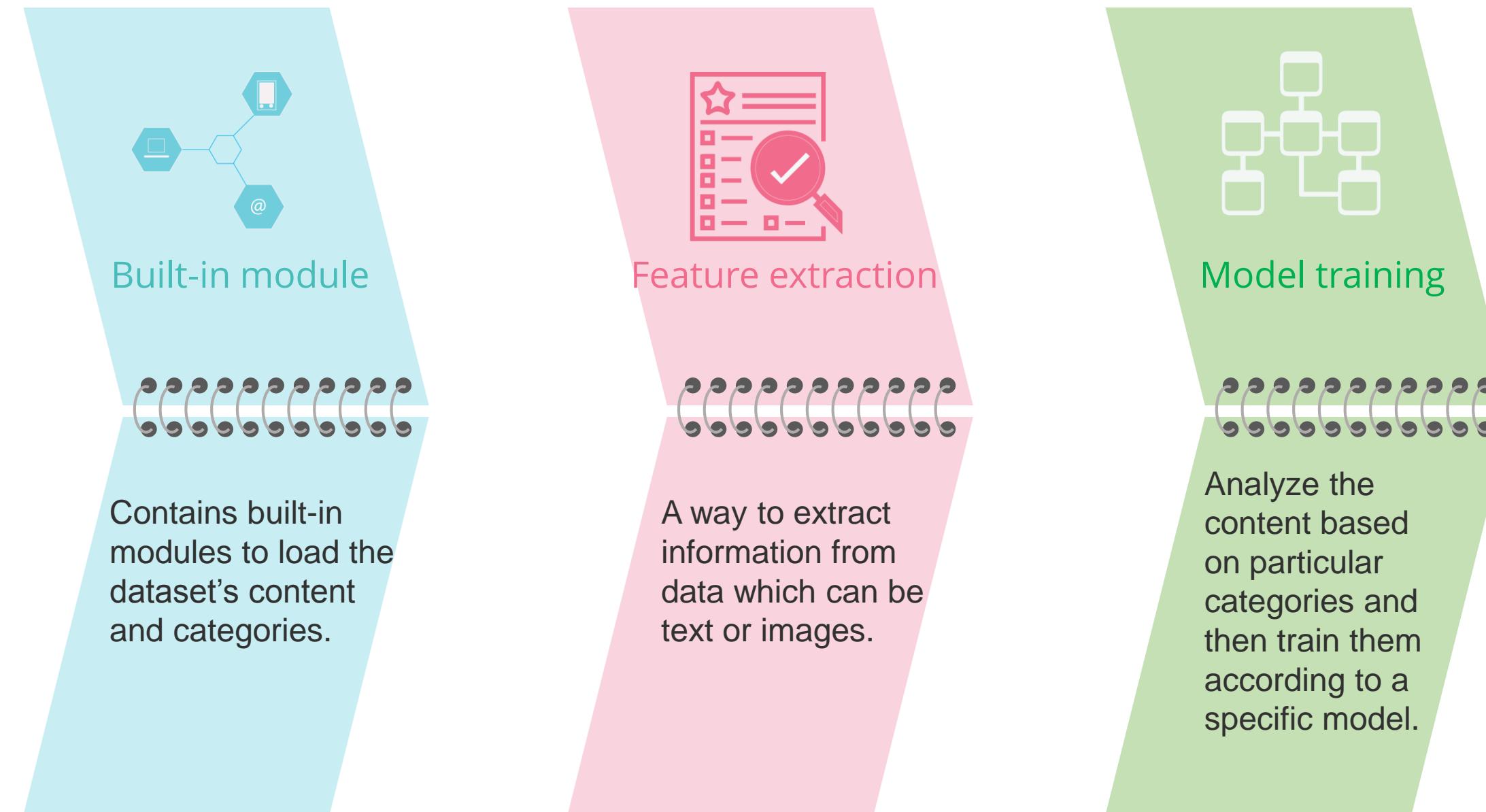
---

The major NLP libraries used in Python are:



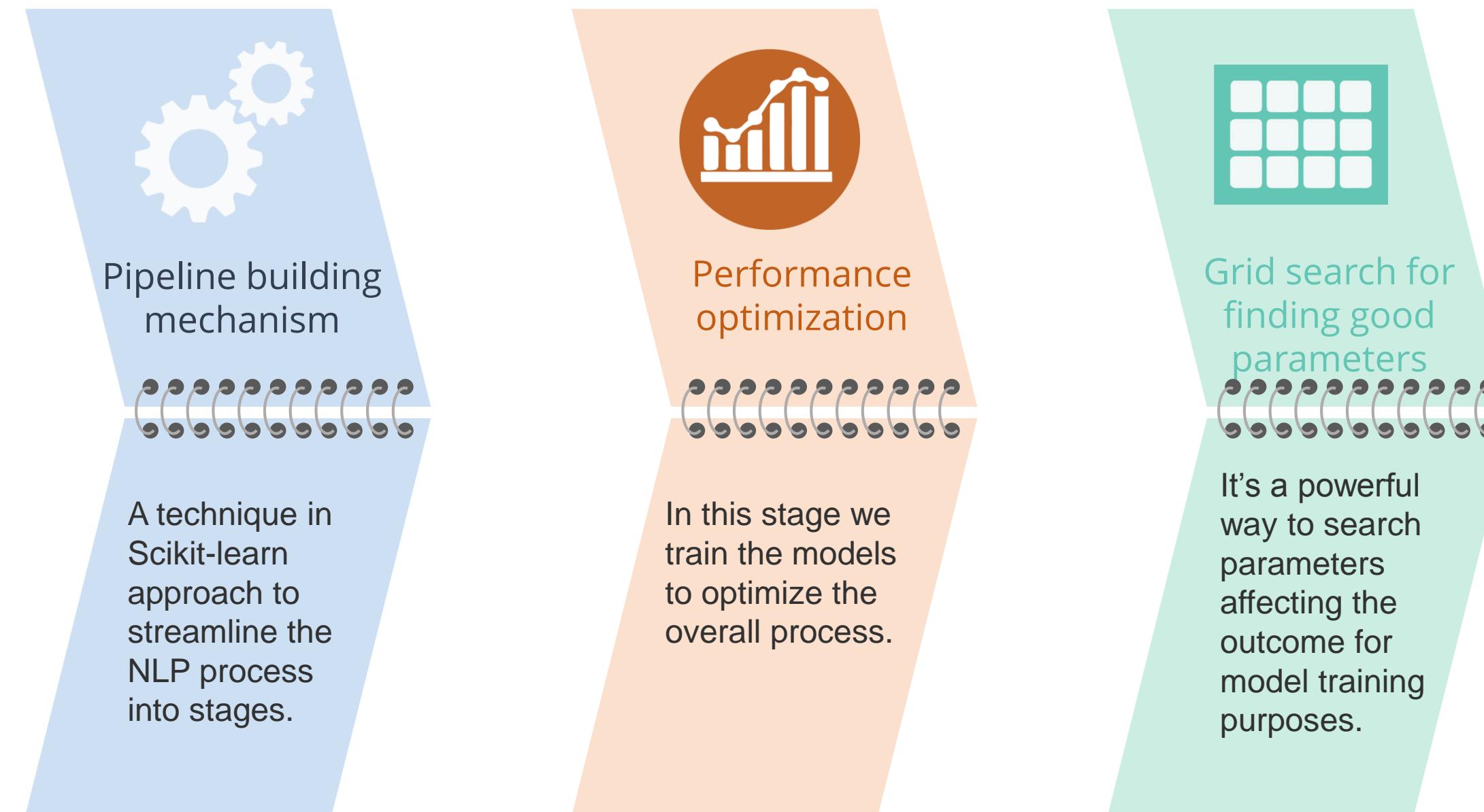
# The Scikit-Learn Approach

It is a very powerful library with a set of modules to process and analyze natural language data such as texts and images and extract information using machine learning algorithms.



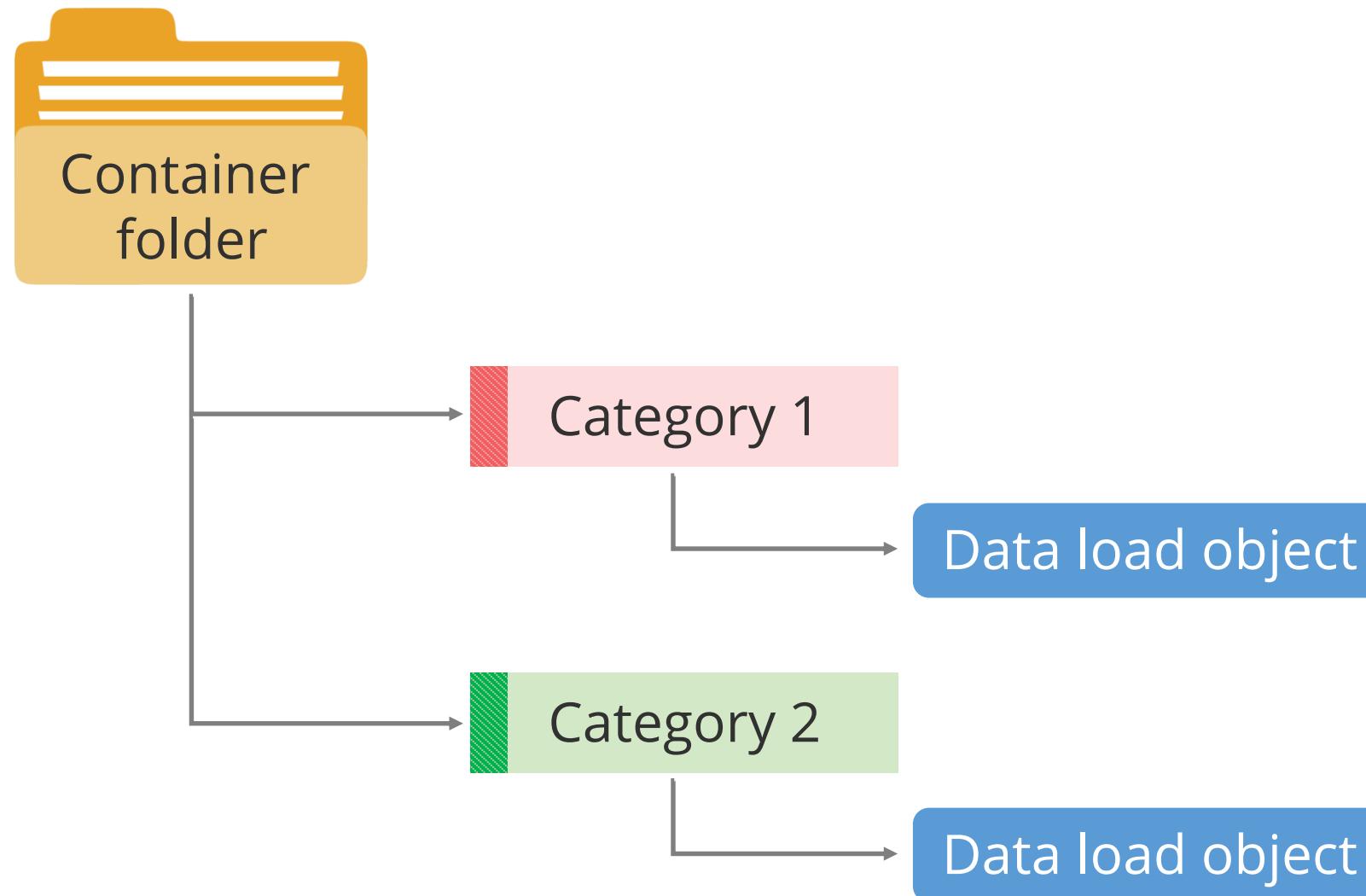
# The SciKit Learn Approach (contd.)

It is a very powerful library with a set of modules to process and analyze natural language data such as texts and images and extract information using machine learning algorithms.



# Modules to Load Content and Category

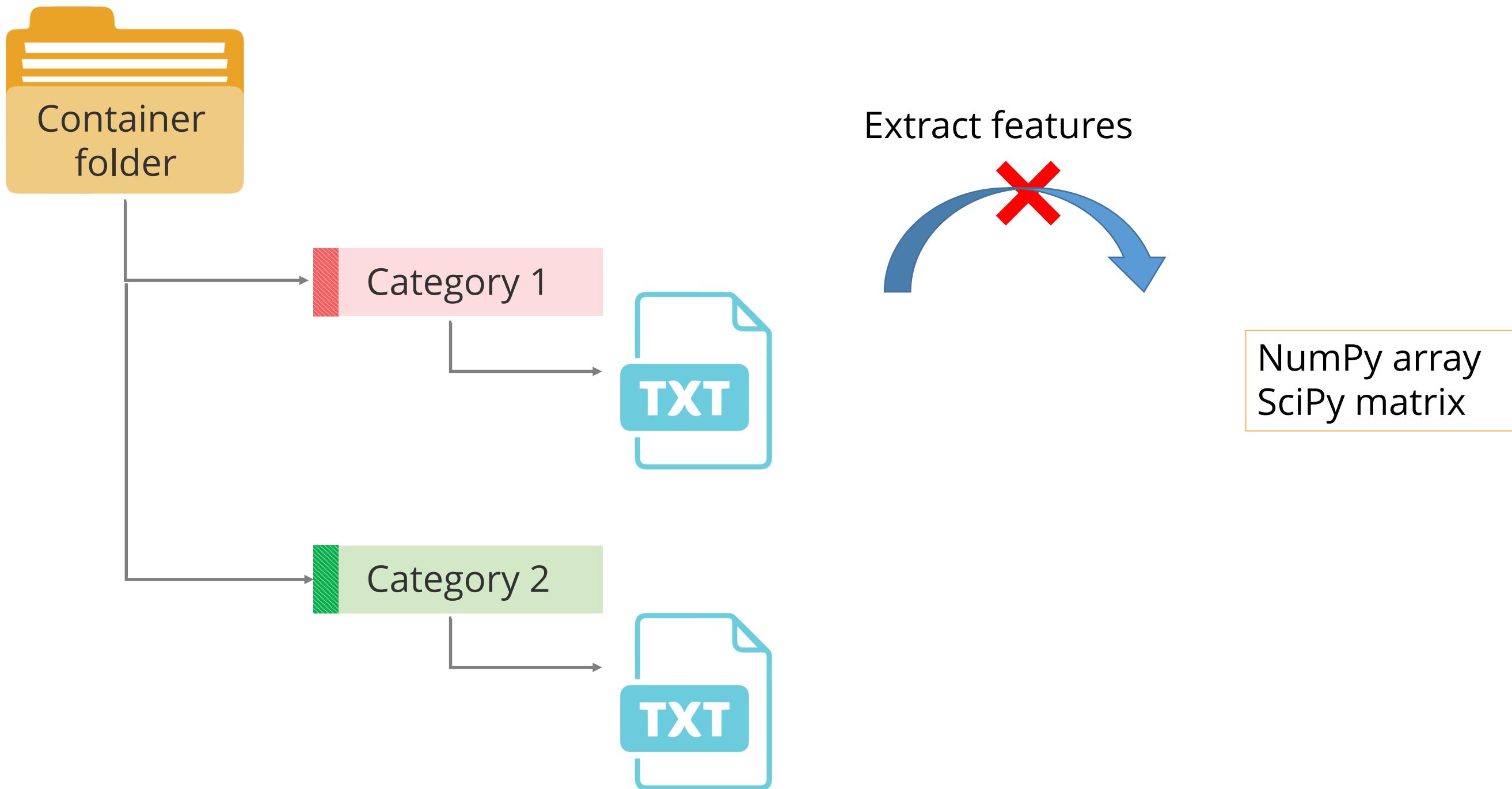
Scikit-learn has many built-in datasets. There are several methods to load these datasets with the help of a data load object.



```
In [ ]: #Load dataset  
load_data = sklearn.datasets.load_files()
```

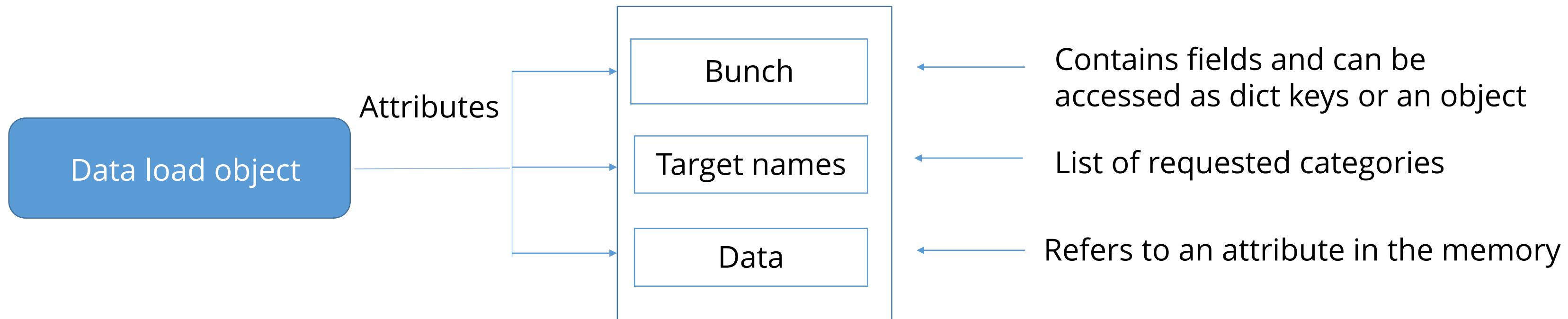
# Modules to Load Content and Category (contd.)

The text files are loaded with categories as subfolder names.



# Modules to Load Content and Category (contd.)

The attributes of a data load object are:



# Modules to Load Content and Category (contd.)

A dataset can be loaded using scikit-learn.

```
In [1]: #Load dataset  
from sklearn.datasets import load_digits
```

Import the dataset

```
In [2]: #create object of the Loaded dataset  
digit_dataset = load_digits()
```

Load dataset

```
In [3]: # use built in descr function to describe dataset  
digit_dataset.DESCR
```

Describe the dataset

```
Out[3]: "Optical Recognition of Handwritten Digits Data Set\n=====  
==\n\nNotes\n----\nData Set Characteristics:\n    :Number of Instances: 5620\n    :Number of Attributes: 64\n    :Attribute Information: 8x8 image of integer pixels in the range 0..16.\n    :Missing Attribute Values: None\n    :Creator: E. Alpaydin (alpaydin '@' boun.edu.tr)\n    :Date: July; 1998\nThis is a copy of the test set of the UCI ML hand-written digits datasets\nhttp://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits\nThe data set contains images of hand-written digits: 10 classes where  
each class refers to a digit.\n\nPreprocessing programs made available by NIST were used to extract  
normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13  
to the test set. 32x32 bitmaps are divided into nonoverlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates  
an input matrix of 8x8 where each element is an integer in the range 16. This produces a
```

# Modules to Load Content and Category (contd.)

Let us see how functions like type, .data, and .target help in analyzing a dataset.

```
In [4]: #view type of dataset  
type(digit_dataset)
```

View type of dataset

```
Out[4]: sklearn.datasets.base.Bunch
```

```
In [5]: #view data  
digit_dataset.data
```

View data

```
Out[5]: array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],  
               [ 0.,  0.,  0., ..., 10.,  0.,  0.],  
               [ 0.,  0.,  0., ..., 16.,  9.,  0.],  
               ...,  
               [ 0.,  0.,  1., ...,  6.,  0.,  0.],  
               [ 0.,  0.,  2., ..., 12.,  0.,  0.],  
               [ 0.,  0., 10., ..., 12.,  1.,  0.]])
```

```
In [6]: #view target  
digit_dataset.target
```

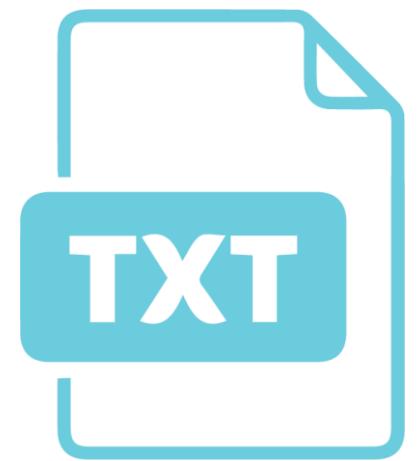
View target

```
Out[6]: array([0, 1, 2, ..., 8, 9, 8])
```

# Feature Extraction

---

Feature extraction is a technique to convert the content into the numerical vectors to perform machine learning.



Text feature extraction



For example: Large datasets or documents

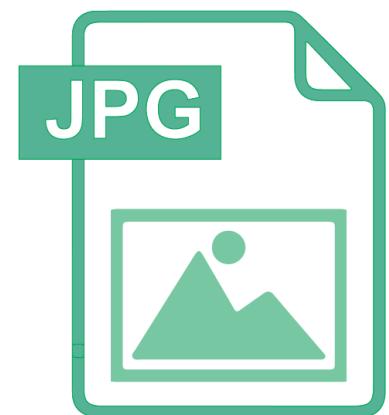


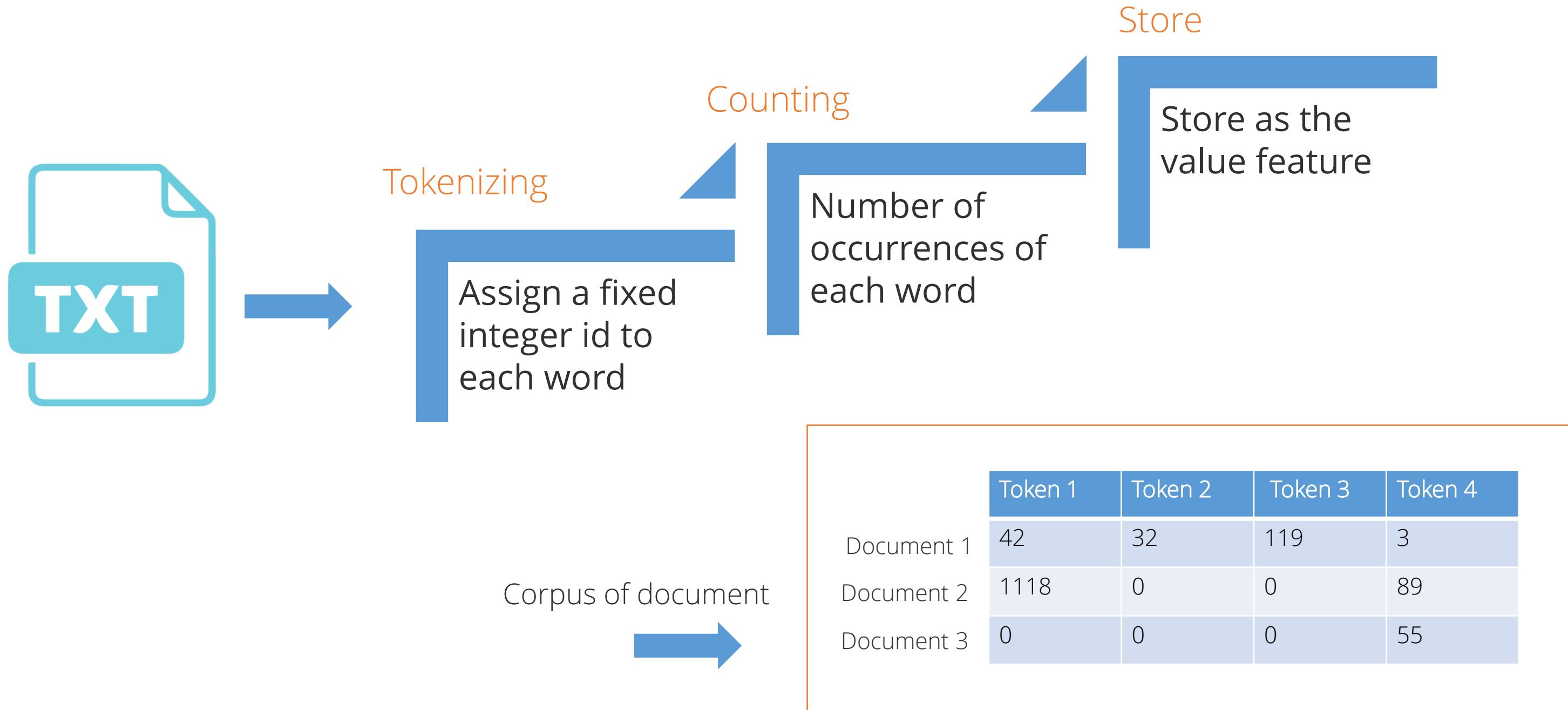
Image feature extraction



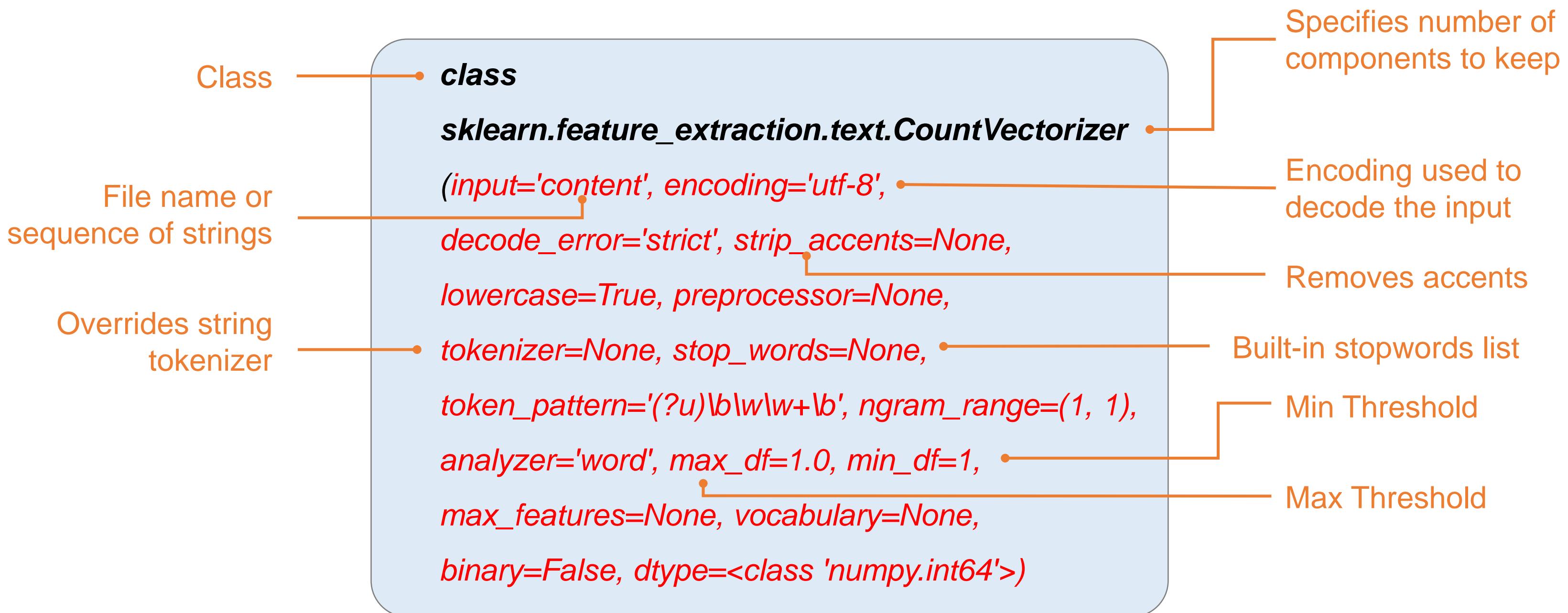
For example: Patch extraction, hierarchical clustering

# Bag of Words

Bag of words is used to convert text data into numerical feature vectors with a fixed size.



# CountVectorizer Class Signature



## Demo 03—Bag of Words

Demonstrate the Bag of Words technique



# Text Feature Extraction Considerations

---

## Sparse

This utility deals with sparse matrix while storing them in memory. Sparse data is commonly noticed when it comes to extracting feature values, especially for large document datasets.

## Vectorizer

It implements tokenization and occurrence. Words with minimum two letters get tokenized. We can use the analyzer function to vectorize the text data.

## Tf-idf

It is a term weighing utility for term frequency and inverse document frequency. Term frequency indicates the frequency of a particular term in the document. Inverse document frequency is a factor which diminishes the weight of terms that occur frequently.

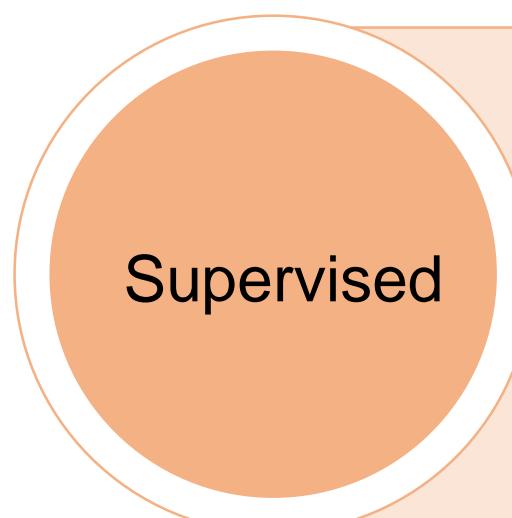
## Decoding

This utility can decode text files if their encoding is specified.

# Model Training

---

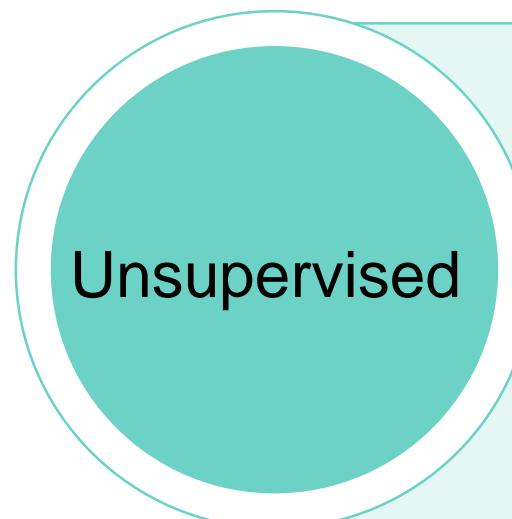
An important task in model training is to identify the right model for the given dataset. The choice of model completely depends on the type of dataset.



Supervised

Models predict the outcome of new observations and datasets, and classify documents based on the features and response of a given dataset.

Example: Naïve Bayes, SVM, linear regression, K-NN neighbors



Unsupervised

Models identify patterns in the data and extract its structure. They are also used to group documents using clustering algorithms.

Example: K-means

# Naïve Bayes Classifier

---

It is the most basic technique for classification of text.

## Advantages:

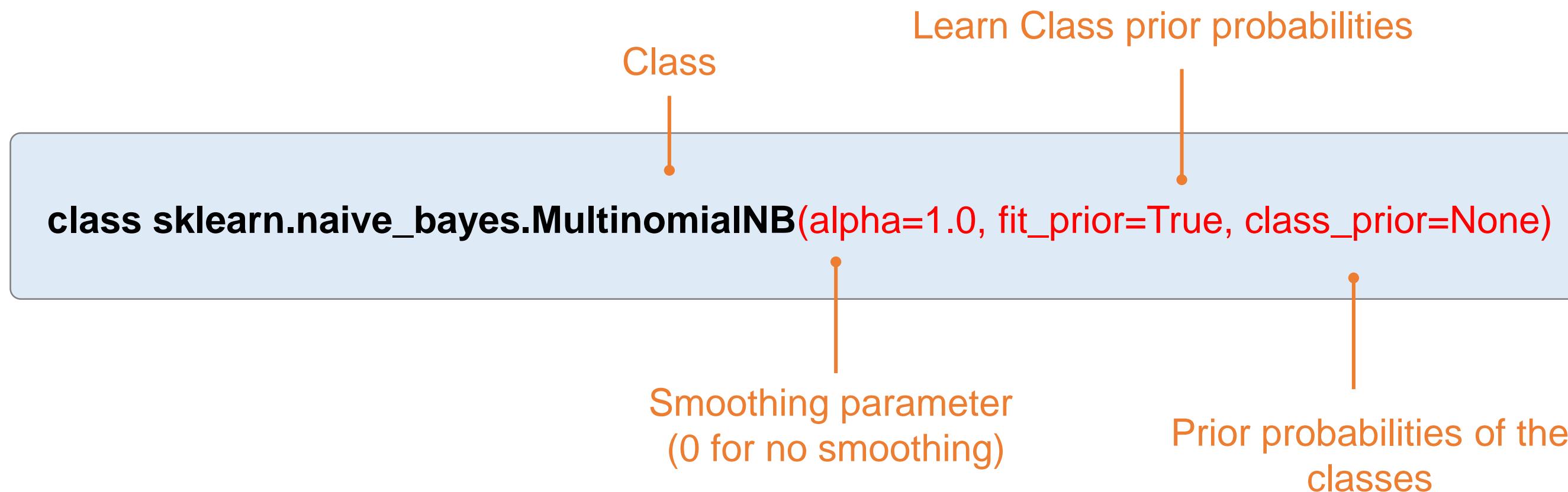
- It is efficient as it uses limited CPU and memory.
- It is fast as the model training takes less time.

## Uses:

- Naïve Bayes is used for sentiment analysis, email spam detection, categorization of documents, and language detection.
- Multinomial Naïve Bayes is used when multiple occurrences of the words matter.

# Naïve Bayes Classifier

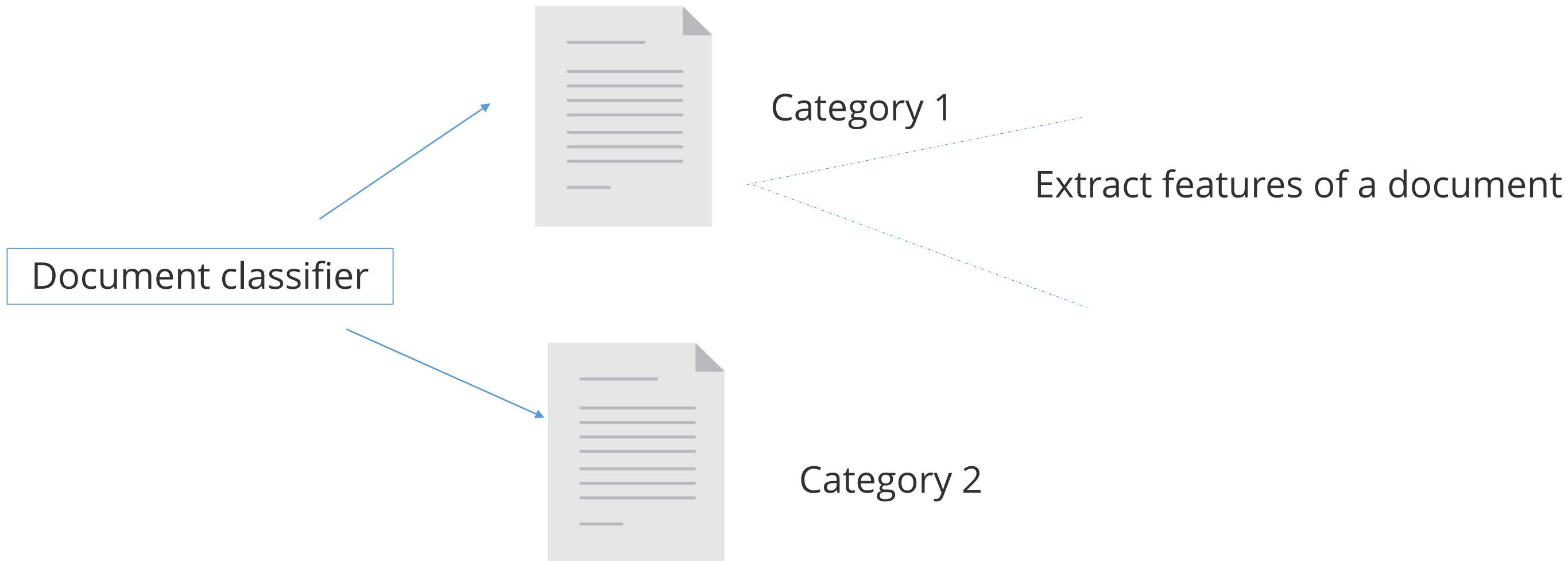
Let us take a look at the signature of the multinomial Naïve Bayes classifier:



# Grid Search and Multiple Parameters

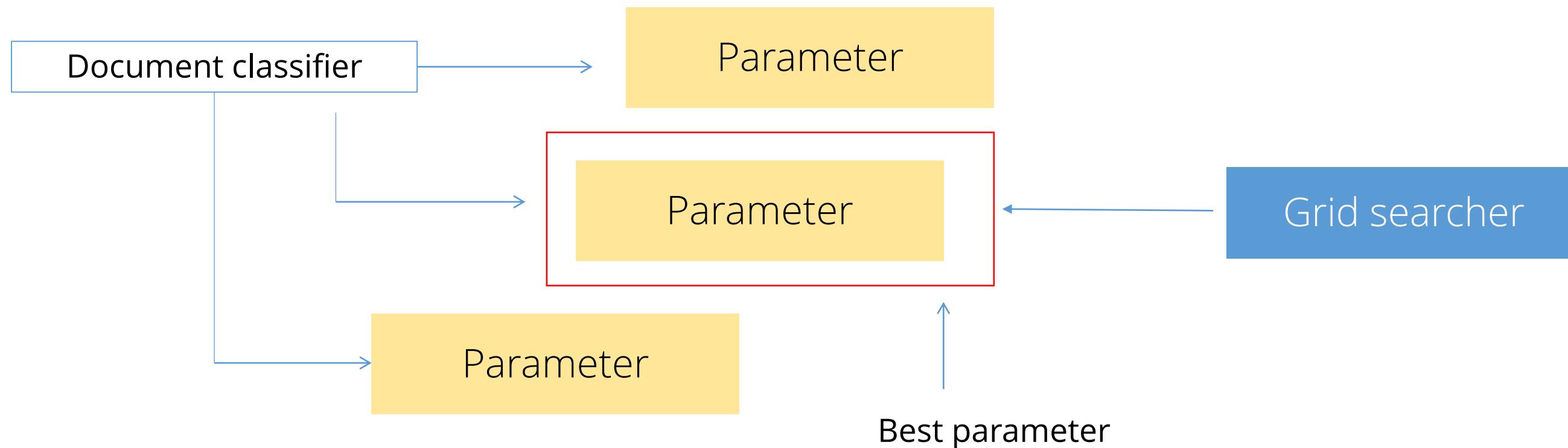
---

Document classifiers can have many parameters and a Grid approach helps to search the best parameters for model training and predicting the outcome accurately.



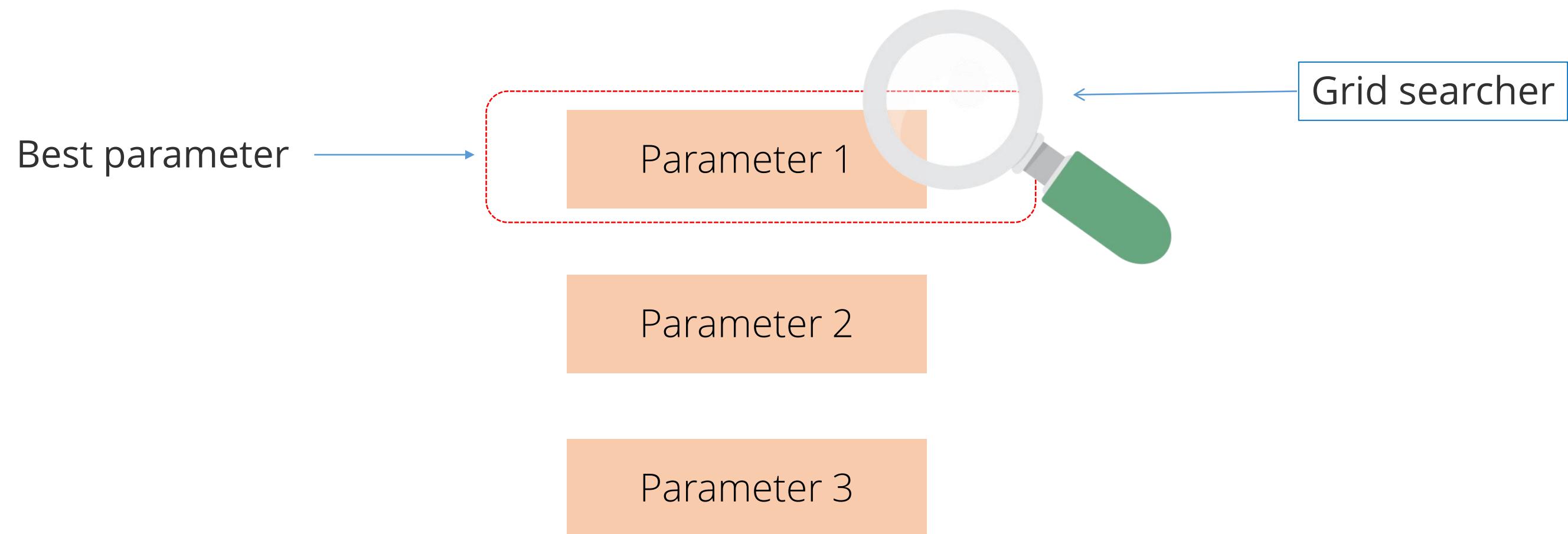
# Grid Search and Multiple Parameters (contd.)

Document classifiers can have many parameters and a Grid approach helps to search the best parameters for model training and predicting the outcome accurately.



# Grid Search and Multiple Parameters (contd.)

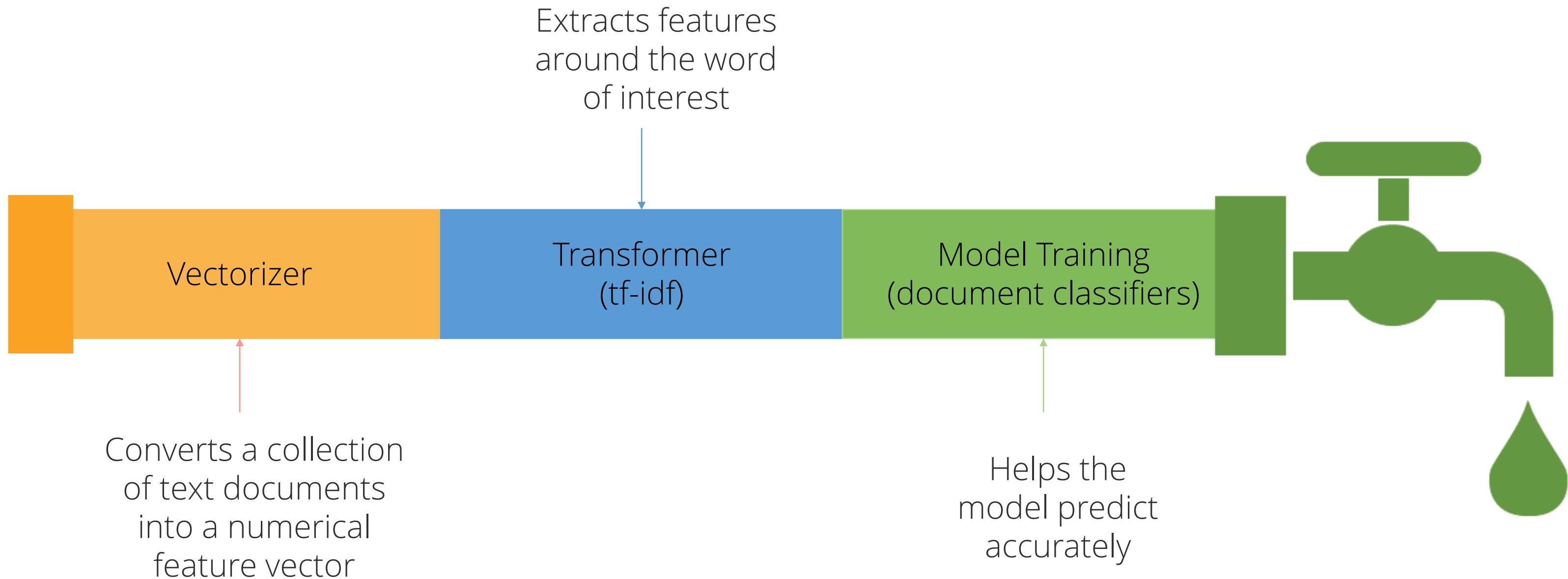
In grid search mechanism, the whole dataset can be divided into multiple grids and a search can be run on entire grids or a combination of grids.



# Pipeline



A pipeline is a combination of vectorizers, transformers, and model training.





## Demo 04—Pipeline and Grid Search

Demonstrate the Pipeline and grid search technique

DATA  
SCIENCE



Problem

Instructions

Analyze the given Spam Collection dataset to:

1. View information on the spam data,
2. View the length of messages,
3. Define a function to eliminate stopwords,
4. Apply Bag of Words,
5. Apply tf-idf transformer, and
6. Detect Spam with Naïve Bayes model.

Problem

Instructions

Instructions on performing the assignment:

- Download the Spam Collection dataset from the “Resource” tab. Upload it using the right syntax to use and analyze it.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



Problem

Assignment

Analyze the Sentiment dataset using NLP to:

1. View the observations,
2. Verify the length of the messages and add it as a new column,
3. Apply a transformer and fit the data in the bag of words,
4. Print the shape for the transformer, and
5. Check the model for predicted and expected values.

Problem

Instructions

Instructions on performing the assignment:

- Download the Sentiment dataset from the “Resource” tab. Upload it to your Jupyter notebook to work on it.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



**QUIZ**

1

**What is the tf-idf value in a document?**

- a. Directly proportional to the number of times a word appears
- b. Inversely proportional to the number of times a word appears
- c. Offset by frequency of the words in corpus
- d. Increase with frequency of the words in corpus



**QUIZ**

1

**What is the tf-idf value in a document?**

- a. Directly proportional to the number of times a word appears
- b. Inversely proportional to the number of times a word appears
- c. Offset by frequency of the words in corpus
- d. Increase with frequency of the words in corpus



The correct answer is

- **a, c**

**Explanation:** td-idf value reflects how important a word is to a document. It is directly proportional to the number of times a word appears and is offset by frequency of the words in corpus.

**QUIZ  
2**

**In grid search if `n_jobs = -1`, then which of the following is correct?**

- a. Uses only 1 CPU core
- b. Detects all installed cores and uses them all
- c. Searches for only one parameter
- d. All parameters will be searched on a given grid



**QUIZ  
2**

**In grid search if `n_jobs = -1`, then which of the following is correct?**

- a. Uses only 1 CPU core
- b. Detects all installed cores and uses them all
- c. Searches for only one parameter
- d. All parameters will be searched on a given grid



The correct answer is . **b**

**Explanation:** Detects all installed cores on the machine and uses all of them.

**QUIZ  
3**

**Identify the correct example of Topic Modeling from the following options:**

- a. Machine translation
- b. Speech recognition
- c. News aggregators
- d. Sentiment analysis



**QUIZ  
3**

**Identify the correct example of Topic Modeling from the following options:**

- a. Machine translation
- b. Speech recognition
- c. News aggregators
- d. Sentiment analysis



The correct answer is . **c**

**Explanation:** 'Topic model' is statistical modeling and used to find latent groupings in the documents based upon the words. For example, news aggregators.

**QUIZ**  
**4**

**How do we save memory while operating on Bag of Words which typically contain high-dimensional sparse datasets?**

- a. Distribute datasets in several blocks or chunks
- b. Store only non zero parts of the feature vectors
- c. Flatten the dataset
- d. Decode them



**QUIZ**  
**4****How do we save memory while operating on Bag of Words which typically contain high-dimensional sparse datasets?**

- a. Distribute datasets in several blocks or chunks
- b. Store only non zero parts of the feature vectors
- c. Flatten the dataset
- d. Decode them



The correct answer is . **b**

**Explanation:** In features vector, there will be several values with zeros. The best way to save memory is to store only non zero parts of the feature vectors.

**QUIZ  
5****What is the function of the sub-module feature\_extraction.text.CountVectorizer?**

- a. Convert a collection of text documents to a matrix of token counts
- b. Convert a collection of text documents to a matrix of token occurrences
- c. Transform a count matrix to a normalized form
- d. Convert a collection of raw documents to a matrix of TF-IDF features



**QUIZ  
5****What is the function of the sub-module feature\_extraction.text.CountVectorizer?**

- a. Convert a collection of text documents to a matrix of token counts
- b. Convert a collection of text documents to a matrix of token occurrences
- c. Transform a count matrix to a normalized form
- d. Convert a collection of raw documents to a matrix of TF-IDF features



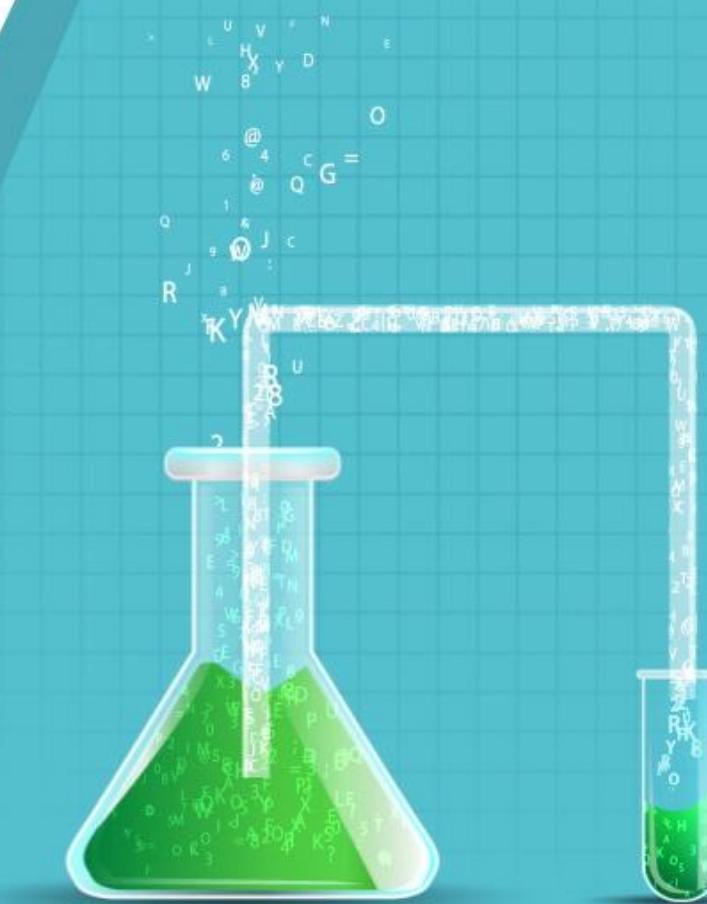
The correct answer is . **a**

**Explanation:** The function of the sub-module feature\_extraction.text.CountVectorizer is to convert a collection of text documents to a matrix of token counts.

## **Key Takeaways**

Let us take a quick recap of what we have learned in the lesson:

- Natural Language Processing is an automated way to understand, analyze human languages, and extract information from such data by applying machine learning algorithms.
- There are various approaches of Natural Language Processing to analyze text data which are inter-dependent or can be independently applied in a document.
- There are two feature extraction techniques, which are text feature extraction and image feature extraction.
- Pipeline building can be used to streamline the NLP process into stages.
- Grid search mechanism is used to perform exhaustive search on the best parameters that impacts the model.



**This concludes 'Natural Language Processing (NLP)  
with Scikit-Learn'**

The next lesson is 'Data Visualization in Python with  
Matplotlib and Bokeh'

DATA  
SCIENCE

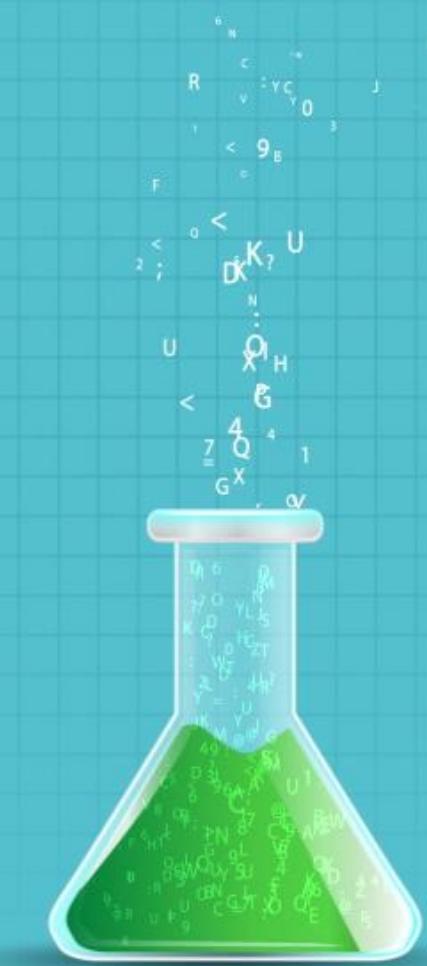
# Data Science with Python

## Lesson 10—Data Visualization in Python using matplotlib



# What's In It For Me

- Explain what data visualization is and its importance in our world today
- Understand why Python is considered one of the best data visualization tools
- Describe matplotlib and its data visualization features in Python
- List the types of plots and the steps involved in creating these plots



# Data Visualization

Data visualization is a technique to present the data in a pictorial or graphical format.

Well, you might wonder why data visualization is important?



## Data Visualization (contd.)

You are a Sales Manager in a leading global organization. The organization plans to study the sales details of each product across all regions and countries. This is to identify the product which has the highest sales in a particular region and up the production. This research will enable the organization to increase the manufacturing of that product in that particular region.



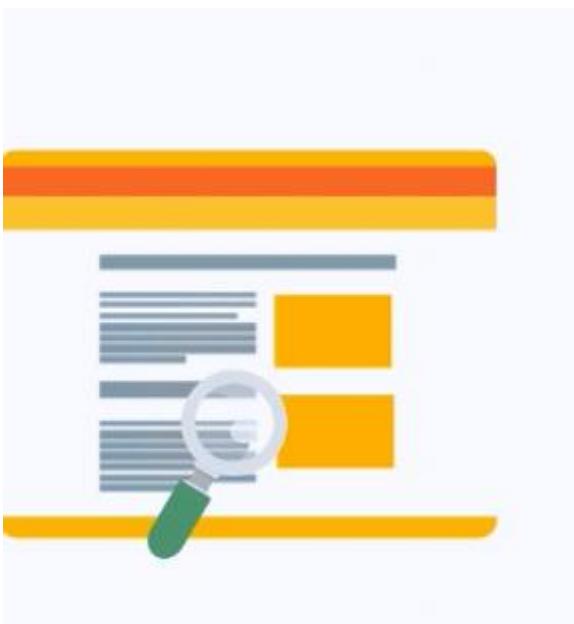
## Data Visualization (contd.)

You are a Sales Manager in a leading global organization. The organization plans to study the sales details of each product across all regions and countries. This is to identify the product which has the highest sales in a particular region and up the production. This research will enable the organization to increase the manufacturing of that product in that particular region.



## Data Visualization (contd.)

You are a Sales Manager in a leading global organization. The organization plans to study the sales details of each product across all regions and countries. This is to identify the product which has the highest sales in a particular region and up the production. This research will enable the organization to increase the manufacturing of that product in that particular region.



# Data Visualization

The main benefits of data visualization are as follows:



# Data Visualization Considerations

Three major considerations for data visualization:



Clarity



Accuracy



Efficiency

Ensure the dataset is complete and relevant. This enables the Data Scientist to use the new patterns obtained from the data in the relevant places.

# Data Visualization Considerations (contd.)

Three major considerations for data visualization:



Clarity



Accuracy



Efficiency

Ensure you use appropriate graphical representation to convey the intended message.

# Data Visualization Considerations (contd.)

Three major considerations for data visualization:



Clarity



Accuracy

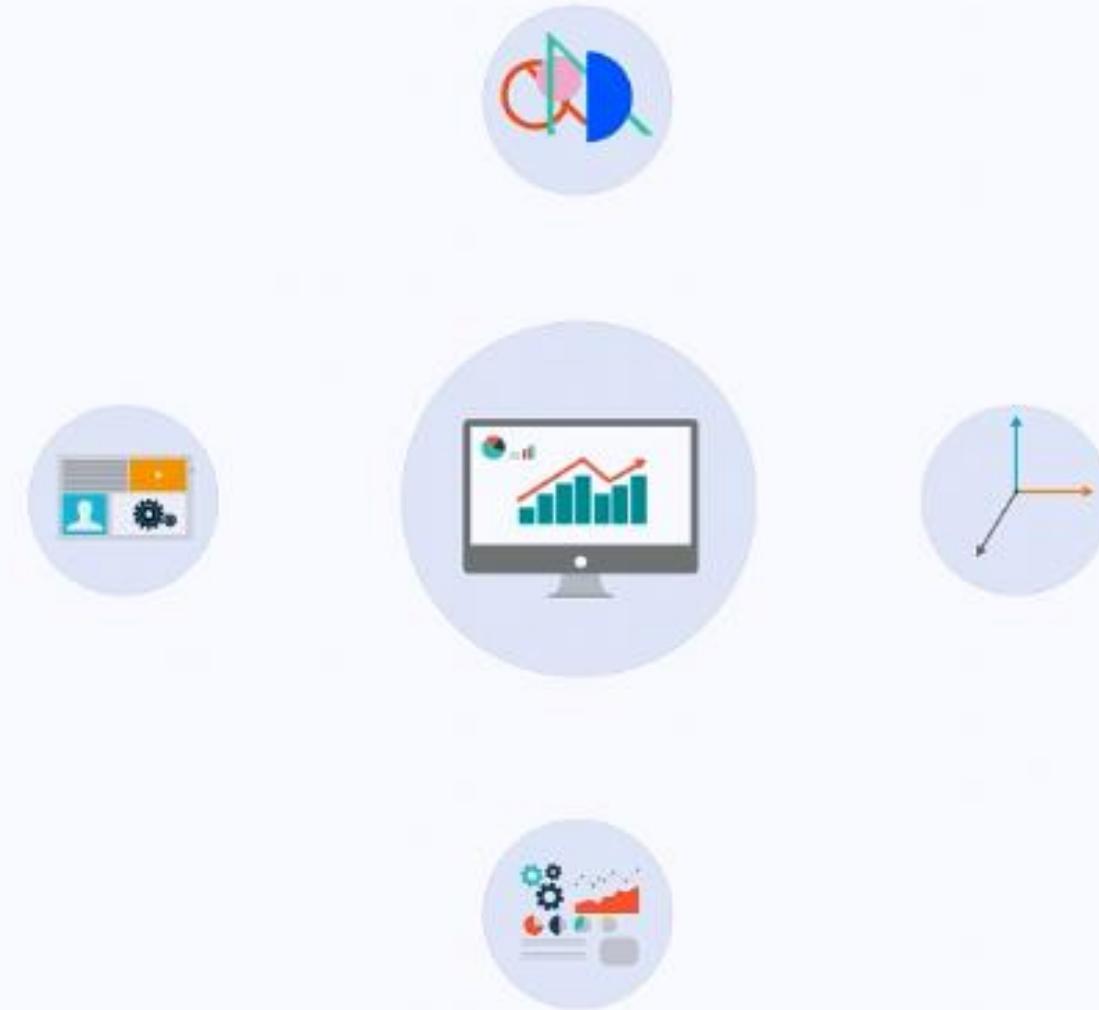


Efficiency

Use efficient visualization techniques that highlight all the data points.

# Data Visualization Factors

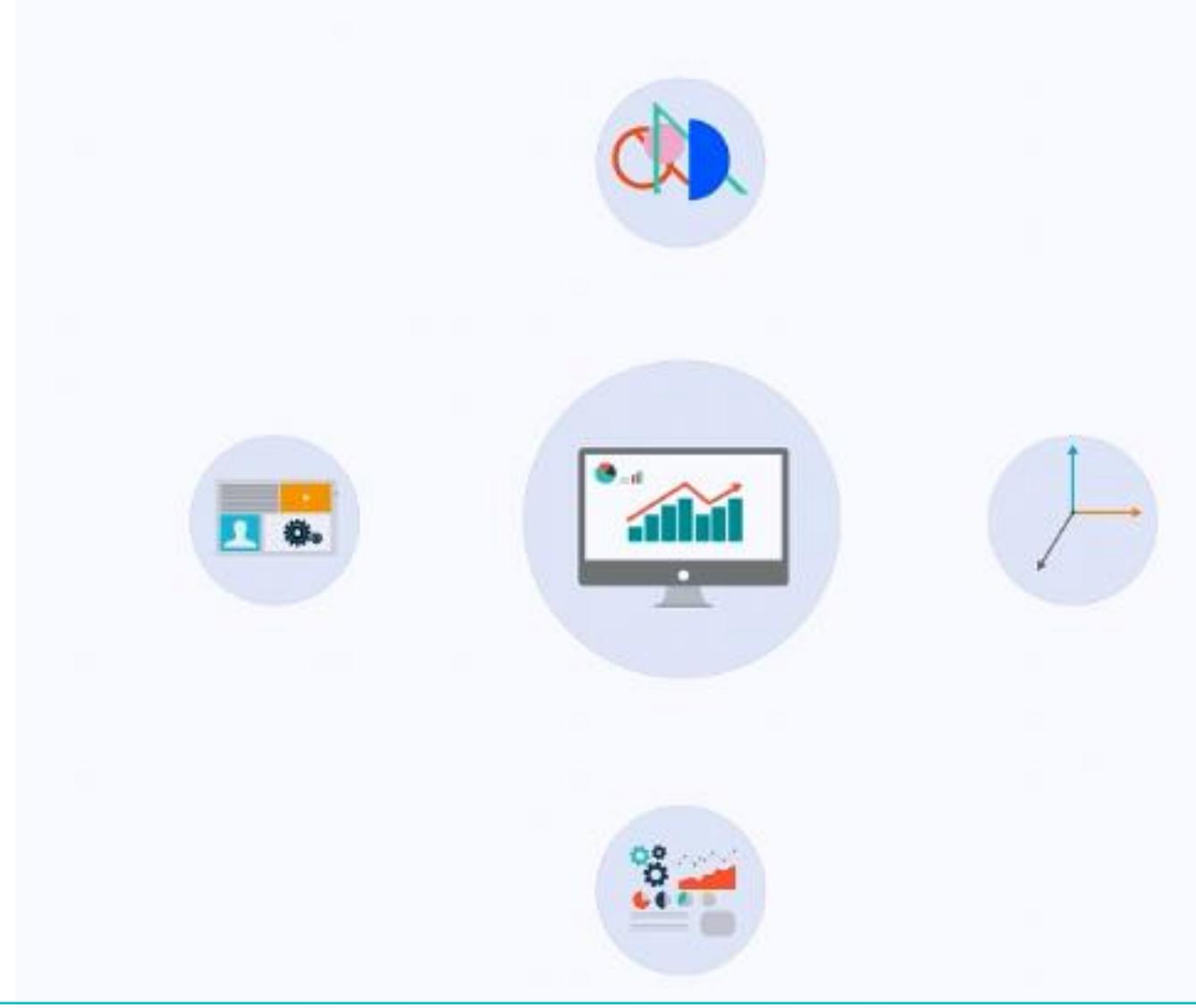
There are some basic factors that one needs to be aware of before visualizing the data:



The visual effect includes the usage of appropriate shapes, colors, and sizes to represent the analyzed data.

## Data Visualization Factors (contd.)

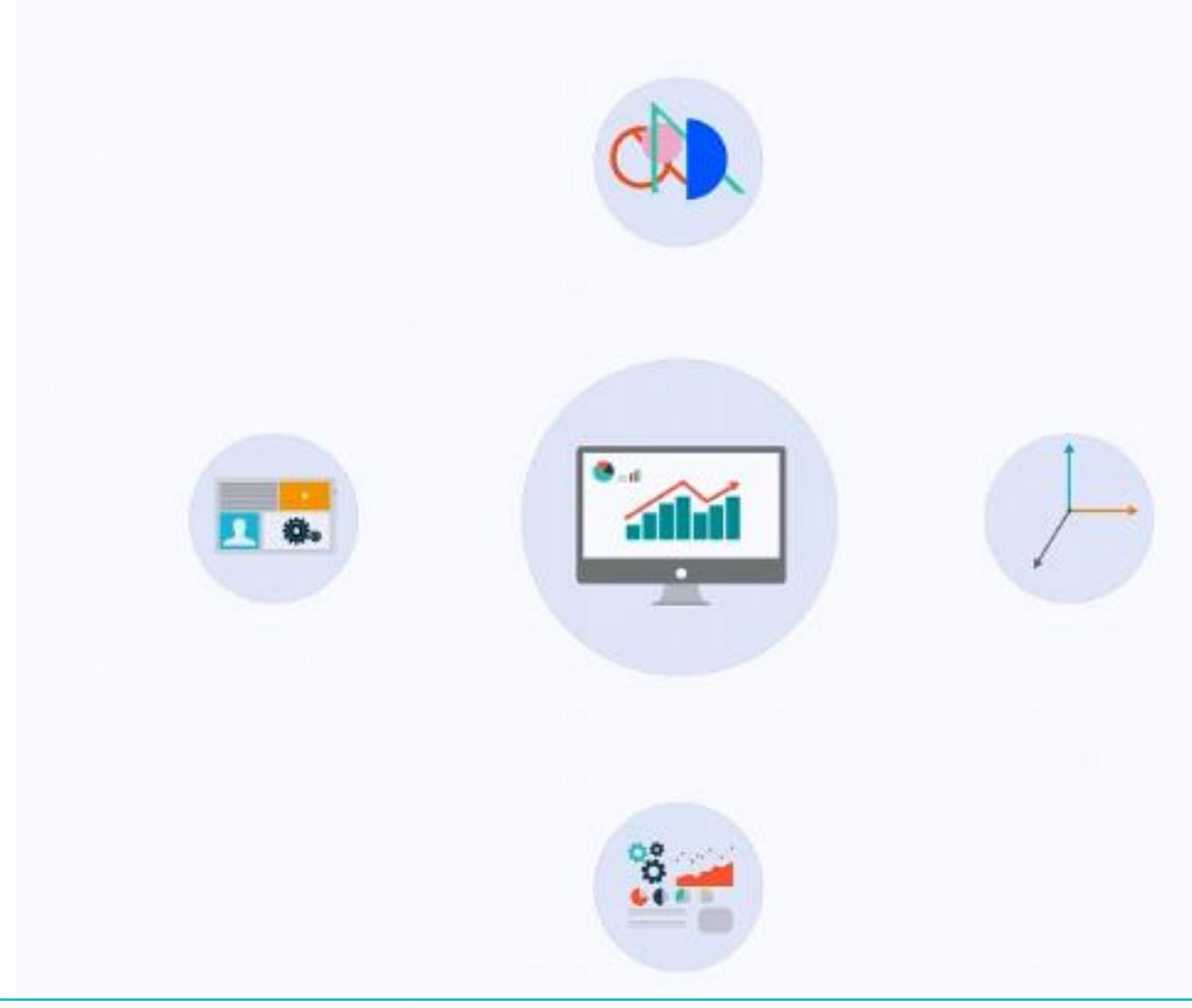
There are some basic factors that one needs to be aware of before visualizing the data:



The coordinate system helps organize the data points within the provided coordinates.

## Data Visualization Factors (contd.)

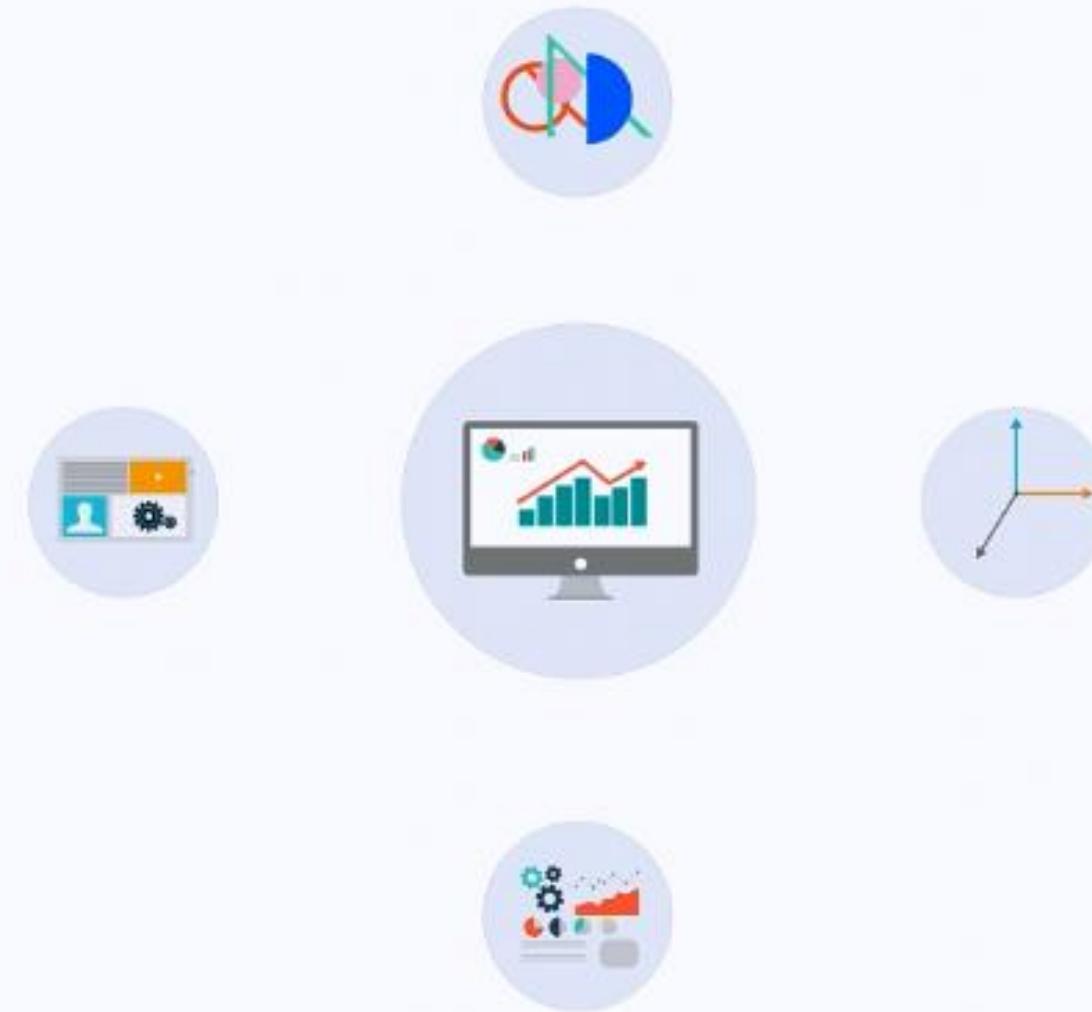
There are some basic factors that one needs to be aware of before visualizing the data:



The data types and scale choose the type of data, for example, numeric or categorical.

# Data Visualization Factors

There are some basic factors that one needs to be aware of before visualizing the data:



The informative interpretation helps create visuals in an effective and easily interpretable manner using labels, title, legends, and pointers.

# Data Visualization Tool—Python

How is data visualization performed for large and complex data?

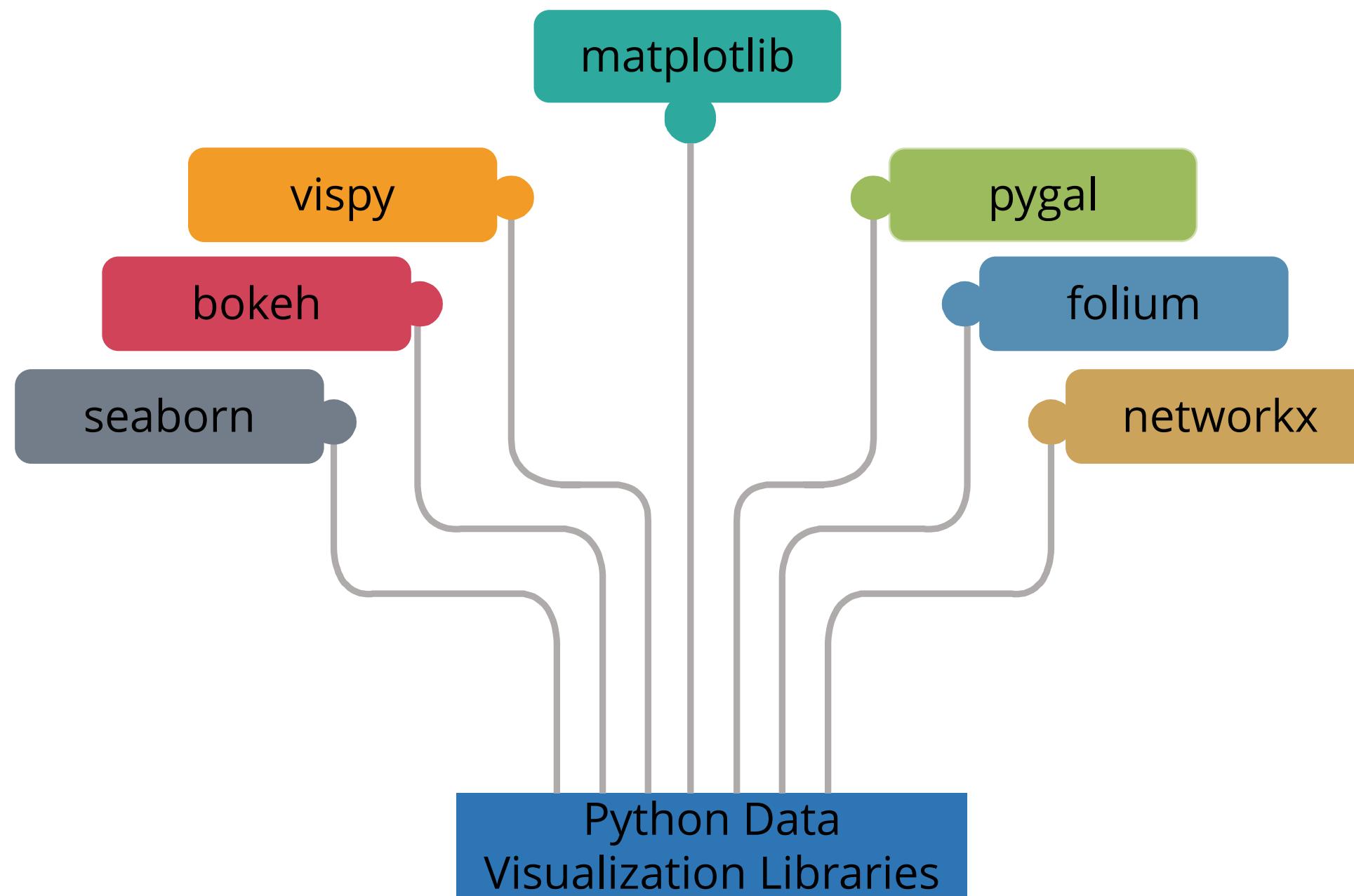


What data visualization is?

How data visualization helps  
interpret results with large data

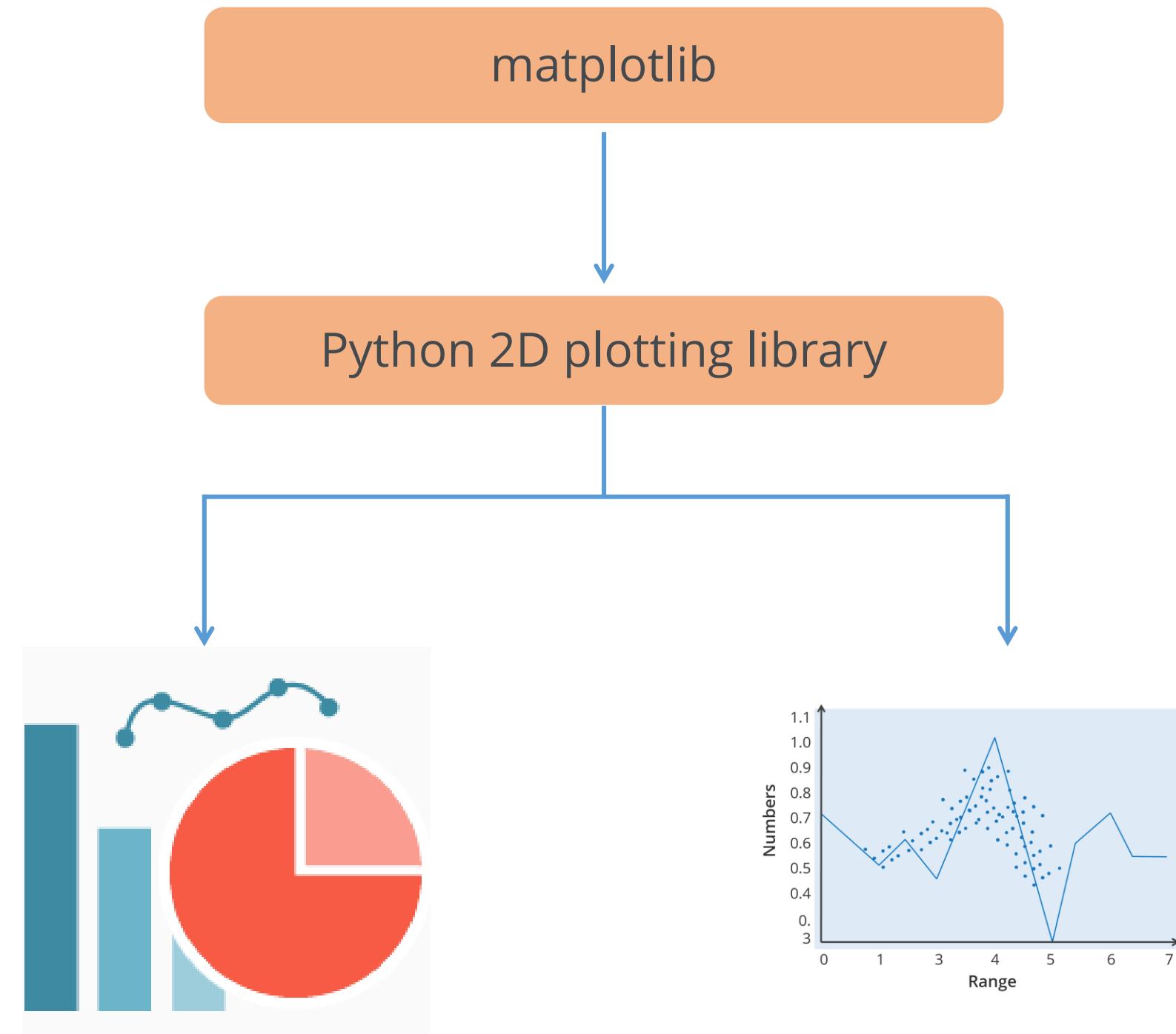
# Python Libraries

Many new Python data visualization libraries are introduced recently such as:



# Python Libraries—matplotlib

Using Python's matplotlib, the data visualization of large and complex data becomes easy.



# Python Libraries—matplotlib (contd.)

There are several advantages of using matplotlib to visualize data. They are as follows:

Is a multi-platform data visualization tool; therefore, it is fast and efficient

Can work well with many operating systems and graphics back ends

Has high-quality graphics and plots to print and view for a range of graphs

With Jupyter notebook integration, the developers are free to spend their time implementing features

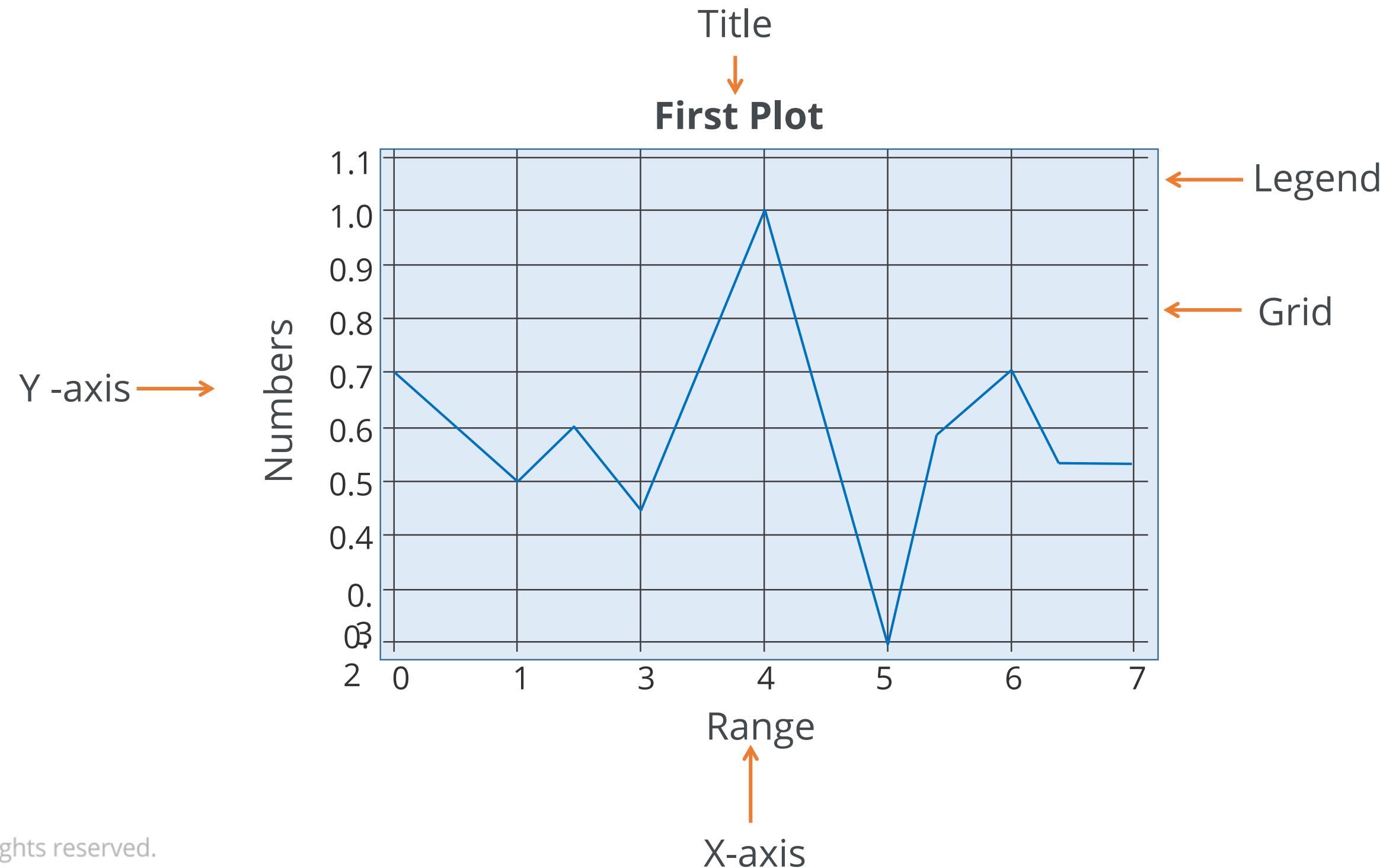
Has large community support and cross platform support as it is an open source tool

Has full control over graphs or plot styles

Advantages of using matplotlib to visualize data

# The Plot

A plot is a graphical representation of data, which shows the relationship between two variables or the distribution of data.



# Steps to Create a Plot

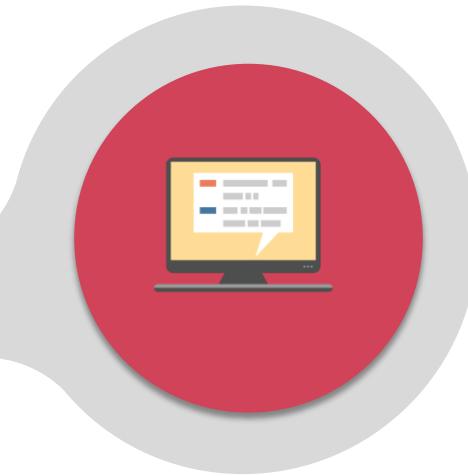
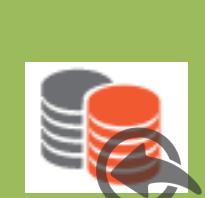
You can create a plot using four simple steps.

Step 01: Import the required libraries



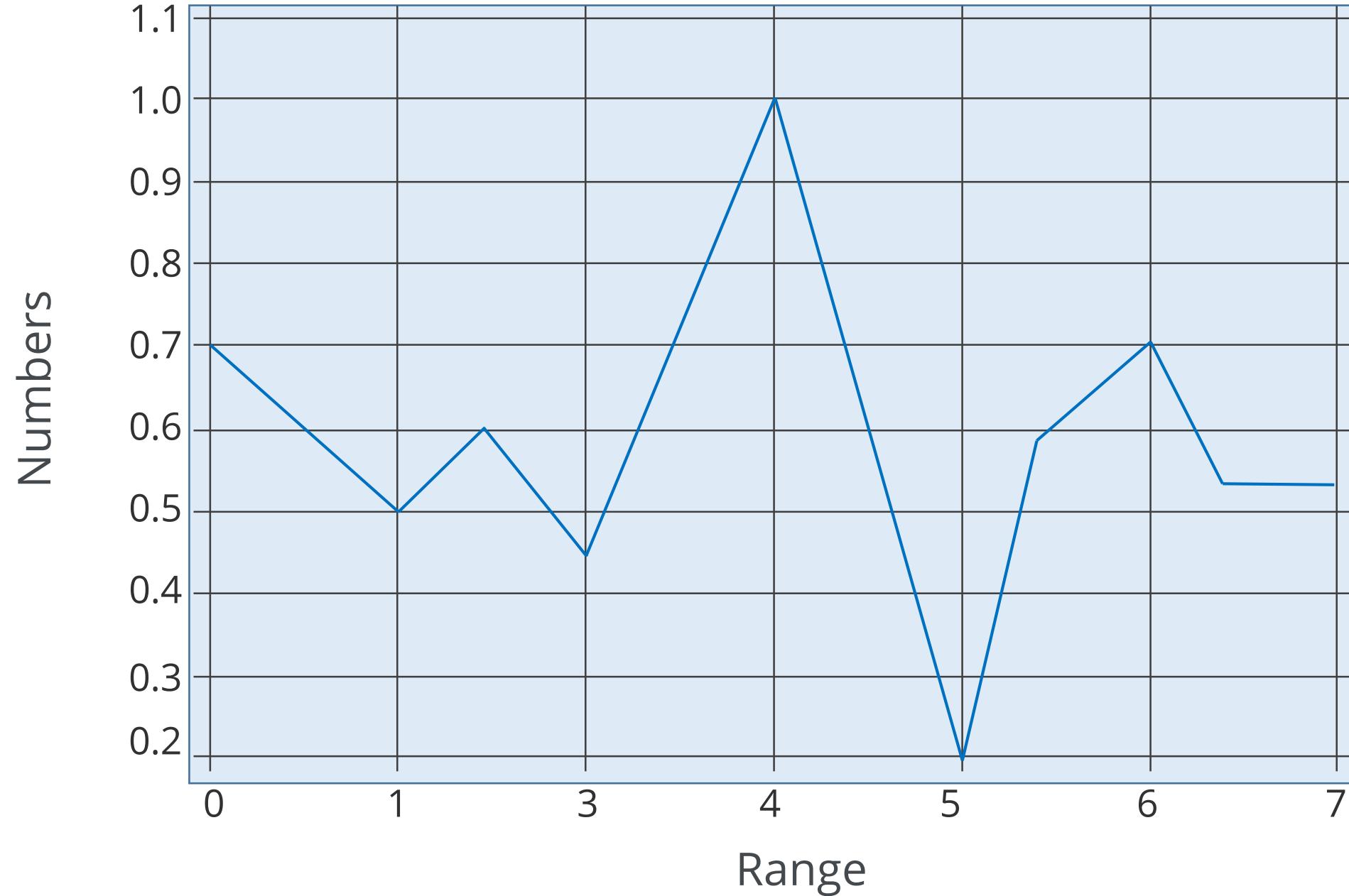
Step 04: Display the created plot

Step 02: Define or import the required dataset



# Steps to Create Plot – Example

First Plot



# Steps to Create Plot – Example (contd.)

```
In [1]: #import numpy for generating random numbers
import numpy as np                                     Generate random numbers      numpy
# import matplotlib library
import matplotlib.pyplot as plt                      Plot the numbers           pyplot
from matplotlib import style                           set the grid style        style
%matplotlib inline
```

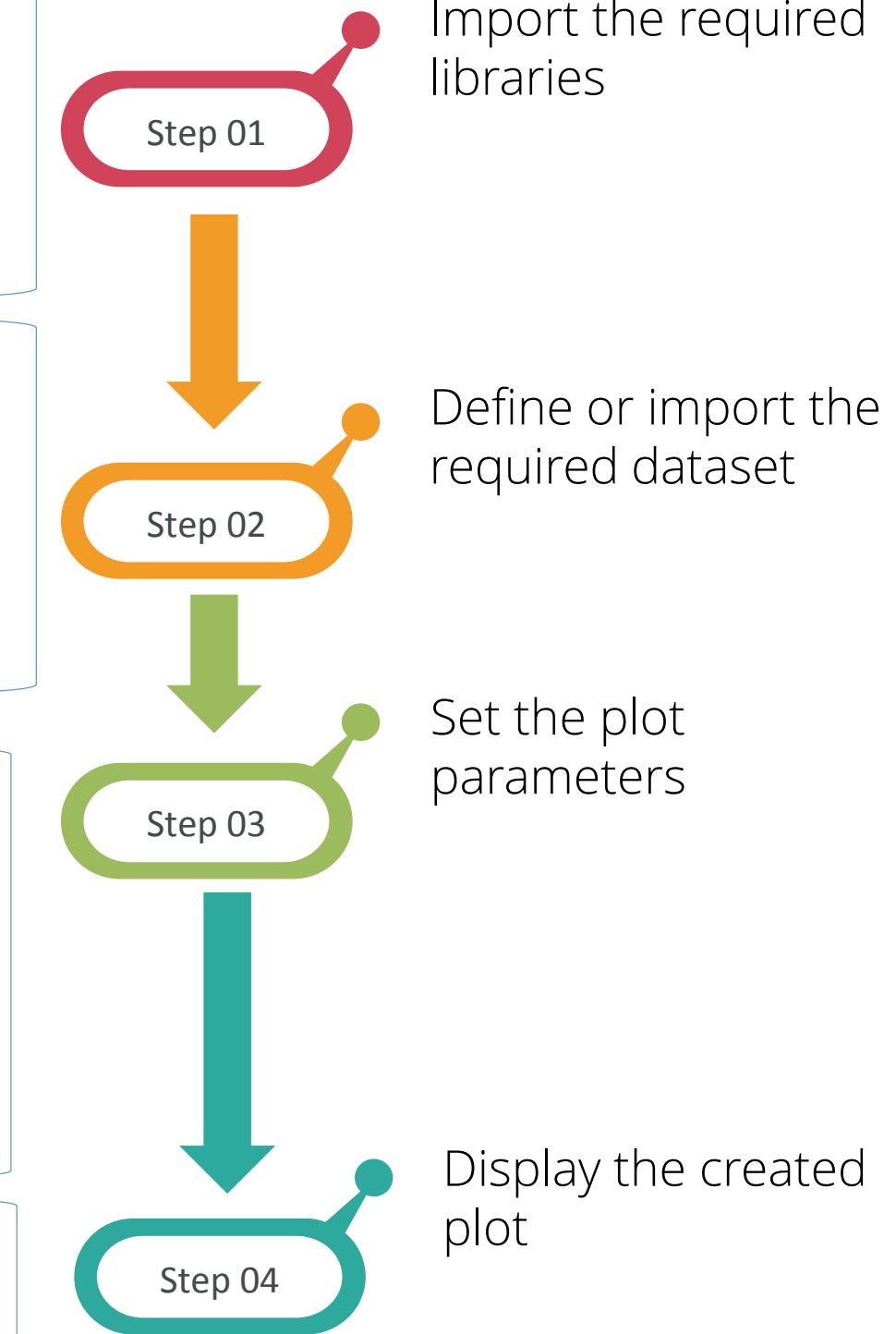
```
In [21]: #generate random numbers (total 10)
randomNumber = np.random.rand(10)                     used numpy random       Defined the dataset
                                                       method
```

```
In [22]: #view them
print randomNumber                                    view the created random
                                                       numbers                  Print method
[ 0.71892609  0.49065612  0.61092193  0.43397501  0.94771363  0.31505178
 0.58568599  0.6929941   0.4288734   0.43774794]
```

```
In [23]: #select the style of the plot
style.use('ggplot')                                  ggplot                  Set the style
#plot the random number
plt.plot(randomNumber,'g',label='line one',linewidth=2) Set the legend
#x axis is number of random numbers (index)
plt.xlabel('Range')                                 Set line width
#y axis is actual random number
plt.ylabel('Numbers')                               Set coordinates labels
#Title of the plot
plt.title('First Plot')                            Set the title
plt.legend()                                       Plot the graph
plt.show()                                         Display the created plot
```





# Knowledge Check

KNOWLEDGE  
CHECK

**Which of the following methods is used to set the title?**

- a. Plot()
- b. Plt.title()
- c. Plot.title()
- d. Title()



KNOWLEDGE  
CHECK**Which of the following methods is used to set the title?**

- a. Plot()
- b. Plt.title()
- c. Plot.title()
- d. Title()



The correct answer is . b.

Explanation Plt.title() is used to set the title.

# Line Properties

## Line Properties

1 alpha

set the transparency  
of the line

2 animated

set the transparency  
of the line

## Plot Graphics

1 linestyle

2 linewidth

3 marker  
style



matplotlib also offers various line colors.

[View Line Properties](#)

***Click View Line Properties to know more.***

# Line Properties (contd.)

Property	Value Type
alpha	float
animated	[True   False]
antialiased or aa	[True   False]
clip_box	a matplotlib.transform.Bbox instance
clip_on	[True   False]
clip_path	a Path instance and a Transform instance, a Patch
color or c	any matplotlib color
contains	the hit testing function
dash_capstyle	['butt'   'round'   'projecting']
linestyle or ls	['-'   '--'   '-.'   ':'   'steps'   ...]
linewidth or lw	float value in points
marker	[ '+'   ','   '.'   '1'   '2'   '3'   '4' ]

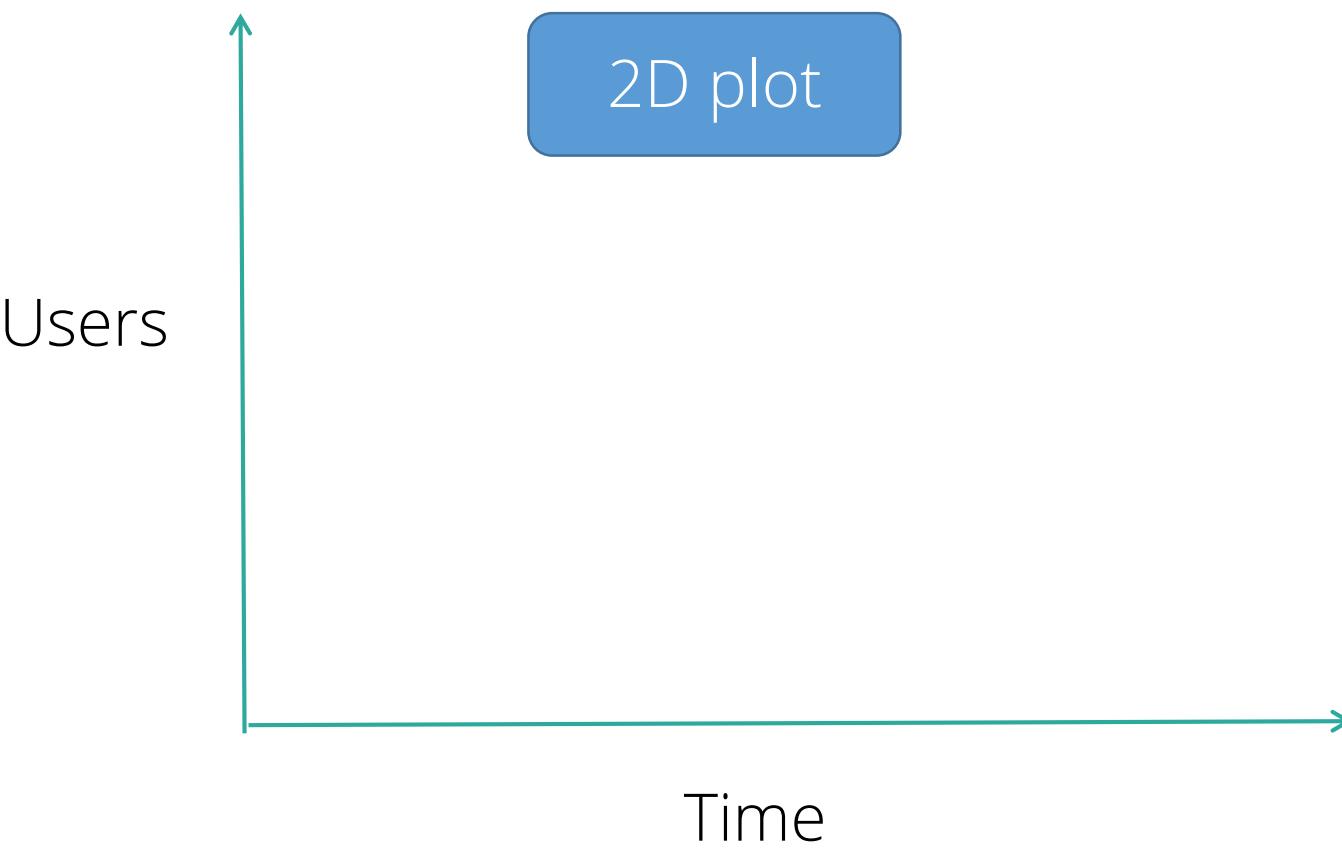
Alias	Color
b	Blue
r	Red
c	Cyan
m	Magenta
g	Green
y	Yellow
k	Black
w	White

[View Line Properties](#)

**Click View Line Properties to know more.**

## Plot With (X,Y)

A leading global organization wants to know how many people visit its website in a particular time. This analysis helps it control and monitor the website traffic.



# Plot With (X,Y)

```
In [1]: #import matplotlib library  
import matplotlib.pyplot as plt  
from matplotlib import style  
%matplotlib inline
```

```
In [2]: #website traffic data  
#number of users/ visitors on the web site  
web_customers = [123,645,950,1290,1630,1450,1034,1295,465,205,80 ]  
#Time distribution (hourly)  
time_hrs = [7,8,9,10,11,12,13,14,15,16,17]
```

List of users

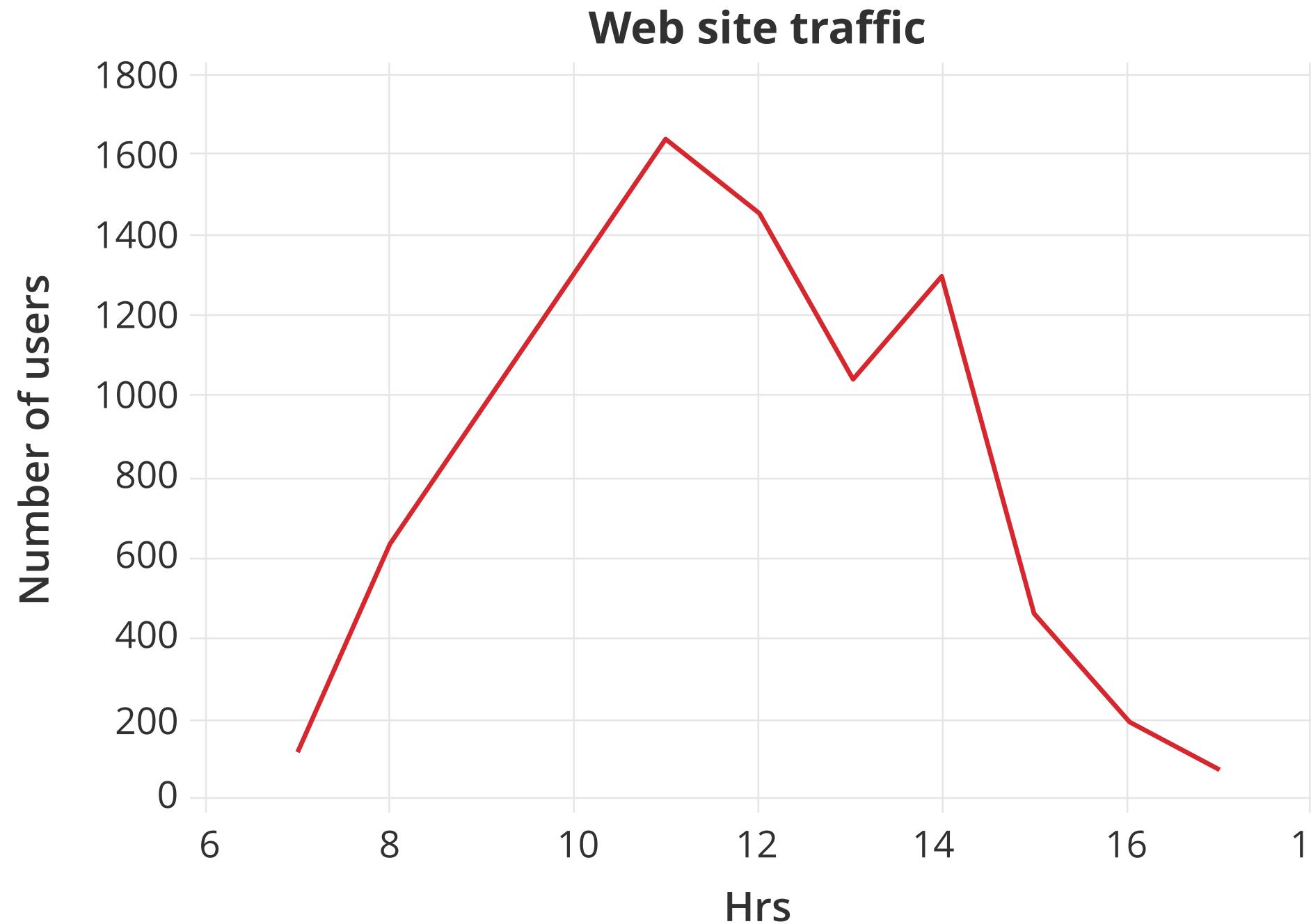
Time

```
In [3]: #select the style of the plot  
style.use('ggplot')  
#plot the web site traffif data (X-axis hrs and Y axis as number of users)  
plt.plot(time_hrs,web_customers)  
#set the title of the plot  
plt.title('Web site traffic')  
#set label for x axis  
plt.xlabel('Hrs')  
#set label for y axis  
plt.ylabel('Number of users')  
plt.show()
```



Use %matplotlib inline to display or view the plot on Jupyter notebook.

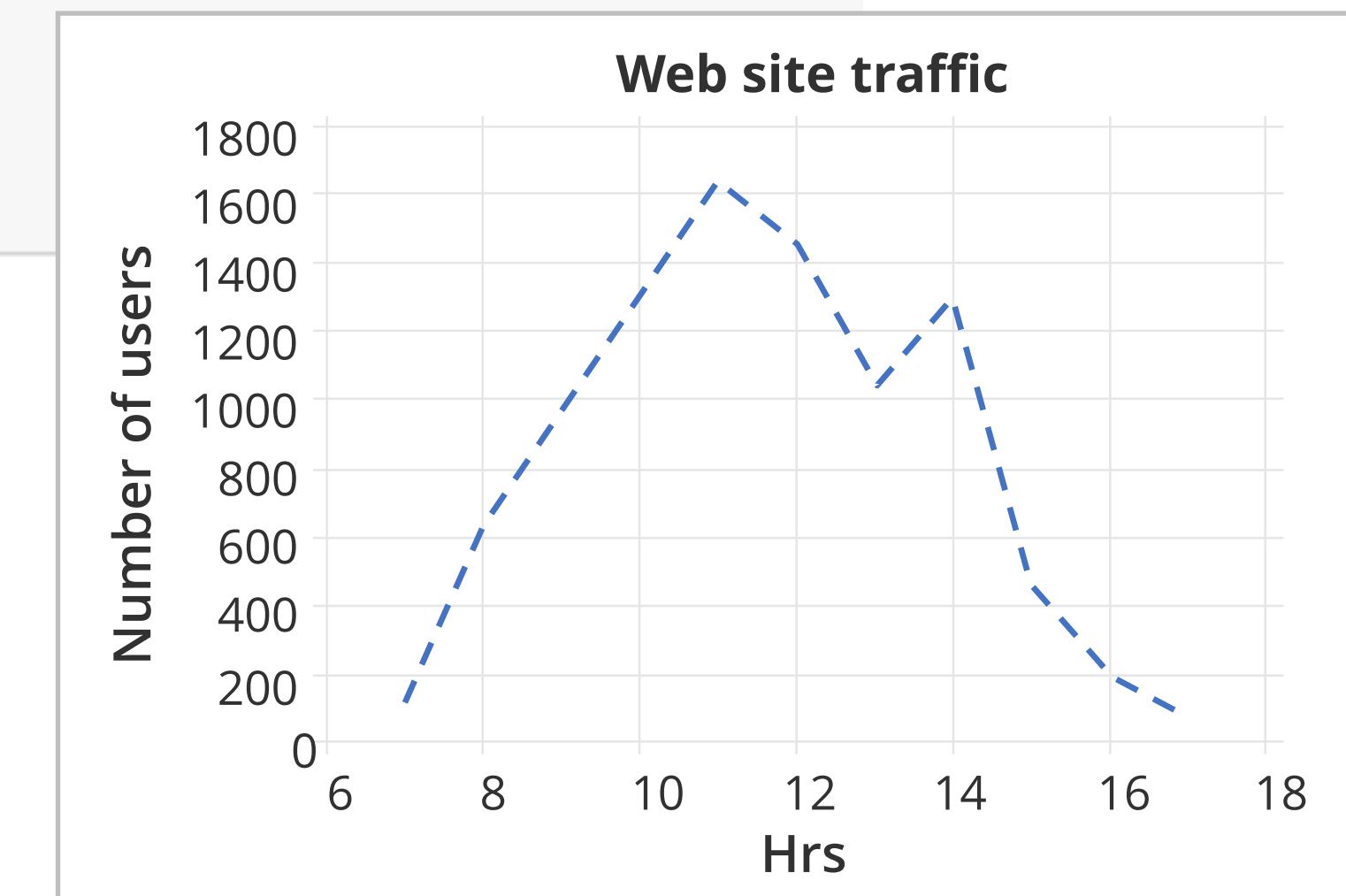
# Plot with (x,y)



# Controlling Line Patterns and Colors

```
#select the style of the plot
style.use('ggplot')
#plot the web stite traffic data (x axis hrs and y asis as number of users)
plt.plot(time_hrs,web_customers,color = 'b',linestyle = '--',linewidth=2.5)
#set the title of the plot
plt.title('Web site traffic')
#set the Label for x axis
plt.xlabel('hrs')
#set the Label for y axis
plt.ylabel('number of users')
plt.show()
```

Line Color (blue)      Dashed (--)

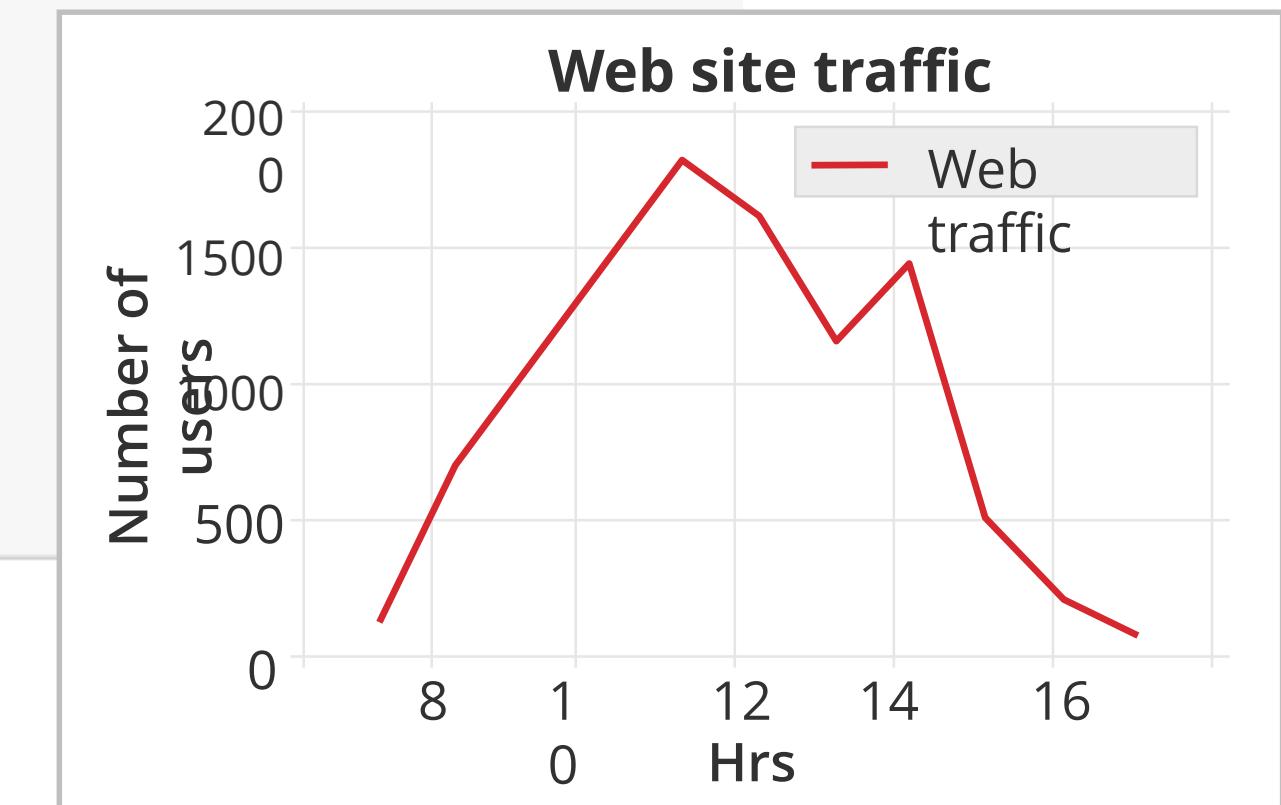


# Set Axis, Labels, and Legend Property

Using matplotlib, it is also possible to set the desired axis to interpret the result.

Axis is used to define the range on the x axis and y axis.

```
: #select the style of the plot
style.use('ggplot')
#plot the web site traffic data (X-axis hrs and Y axis as number of users)
plt.plot(time_hrs,web_customers,'r',label='web traffic',linewidth=1.5)
plt.axis([6.5,17.5,50,2000]) ← Set the axis
#set the title of the plot
plt.title('Web site traffic')
#set label for x axis
plt.xlabel('Hrs')
#set label for y axis
plt.ylabel('Number of users')
plt.legend()
plt.show()
```



# Alpha and Annotation

Alpha is an attribute that controls the transparency of the line.  
The lower the alpha value, the more transparent the line is.

```
#select the style of the plot
style.use('ggplot')
#plot the web stite traffic data (x axis hrs and y asis as number of users)
#also setting the alpha value for transparency
plt.plot(time_hrs,web_customers,alpha=.4)
#set the title of the plot
plt.title('Website Traffic')
#Annotate
plt.annotate('Max',ha='center',va='bottom',xytext=(8,1500),xy=(11,1630),arrowprops =
             { 'facecolor' : 'green'})
#set the Label for x axis
plt.xlabel('hrs')
#set the Label for y axis
plt.ylabel('number of users')

plt.show()
```

# Alpha and Annotation

Annotate() method is used to annotate the graph. It has several attributes which help annotate the plot.

```
#select the style of the plot
style.use('ggplot')
#plot the web site traffic data (x axis hrs and y axis as number of users)
#also setting the alpha value for transparency
plt.plot(time_hrs,web_customers,alpha=.4)
#set the title of the plot
plt.title('Website Traffic')
#Annotate
plt.annotate('Max',ha='center',va='bottom',xytext=(8,1500),xy=(11,1630),arrowprops =
    { 'facecolor' : 'green'})
#set the label for x axis
plt.xlabel('hrs')
#set the label for y axis
plt.ylabel('number of users')

plt.show()
```

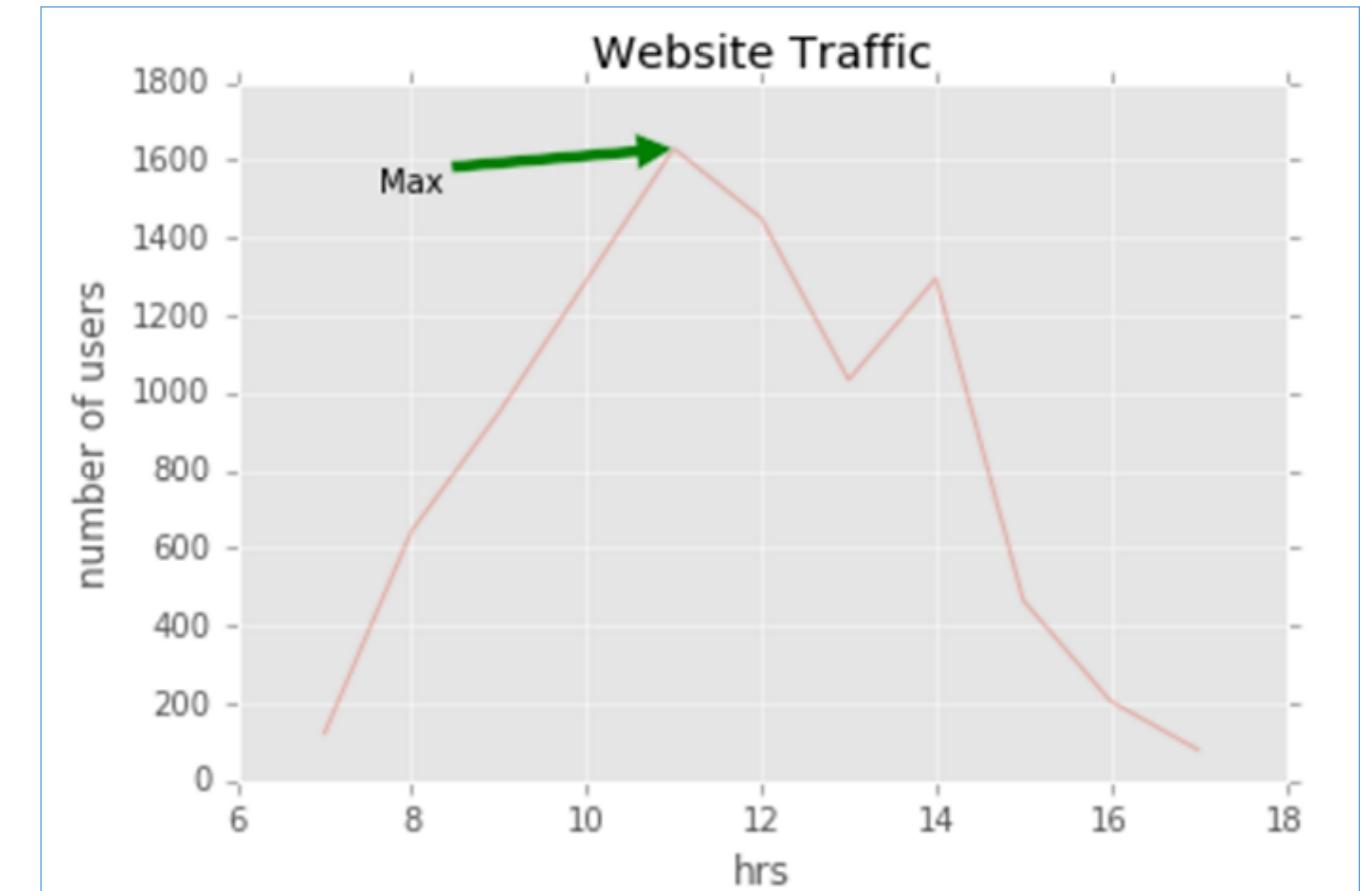
“Max” denotes the annotation text, “ha” indicates the horizontal alignment, “va” indicates the vertical alignment, “xytext” indicates the text position, “xy” indicates the arrow position, and “arrowprops” indicates the properties of the arrow.

# Alpha and Annotation

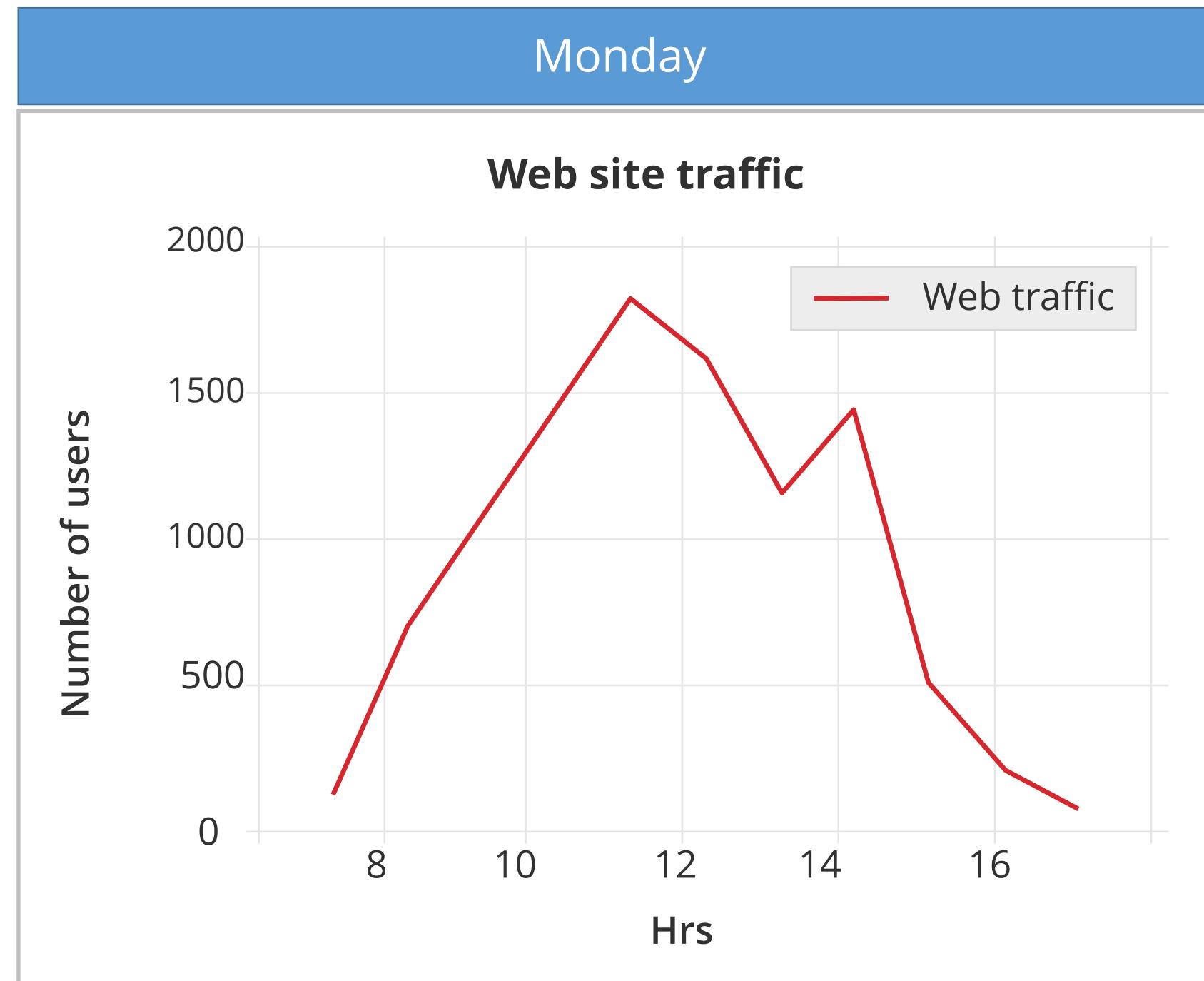
Annotate() method is used to annotate the graph. It has several attributes which help annotate the plot.

```
#select the style of the plot
style.use('ggplot')
#plot the web stite traffic data (x axis hrs and y asis as number of users)
#also setting the alpha value for transparency
plt.plot(time_hrs,web_customers,alpha=.4)
#set the title of the plot
plt.title('Website Traffic')
#Annotate
plt.annotate('Max',ha='center',va='bottom',xytext=(8,1500),xy=(11,1630),arrowprops =
    { 'facecolor' : 'green'})
#set the Label for x axis
plt.xlabel('hrs')
#set the Label for y axis
plt.ylabel('number of users')

plt.show()
```



# Multiple Plots



# Multiple Plots

```
In [4]: #website traffic data  
#number of users/ visitors on the web site  
#monday web traffic  
web_monday = [123,645,950,1290,1630,1450,1034,1295,465,205,80]  
#tuesday web traffic  
web_tuesday= [95,680,889,1145,1670,1323,1119,1265,510,310,110]  
#wednesday web traffic  
web_wednesday= [105,630,700,1006,1520,1124,1239,1380,580,610,230]  
#Time distribution (hourly)  
time_hrs = [7,8,9,10,11,12,13,14,15,16,17]
```

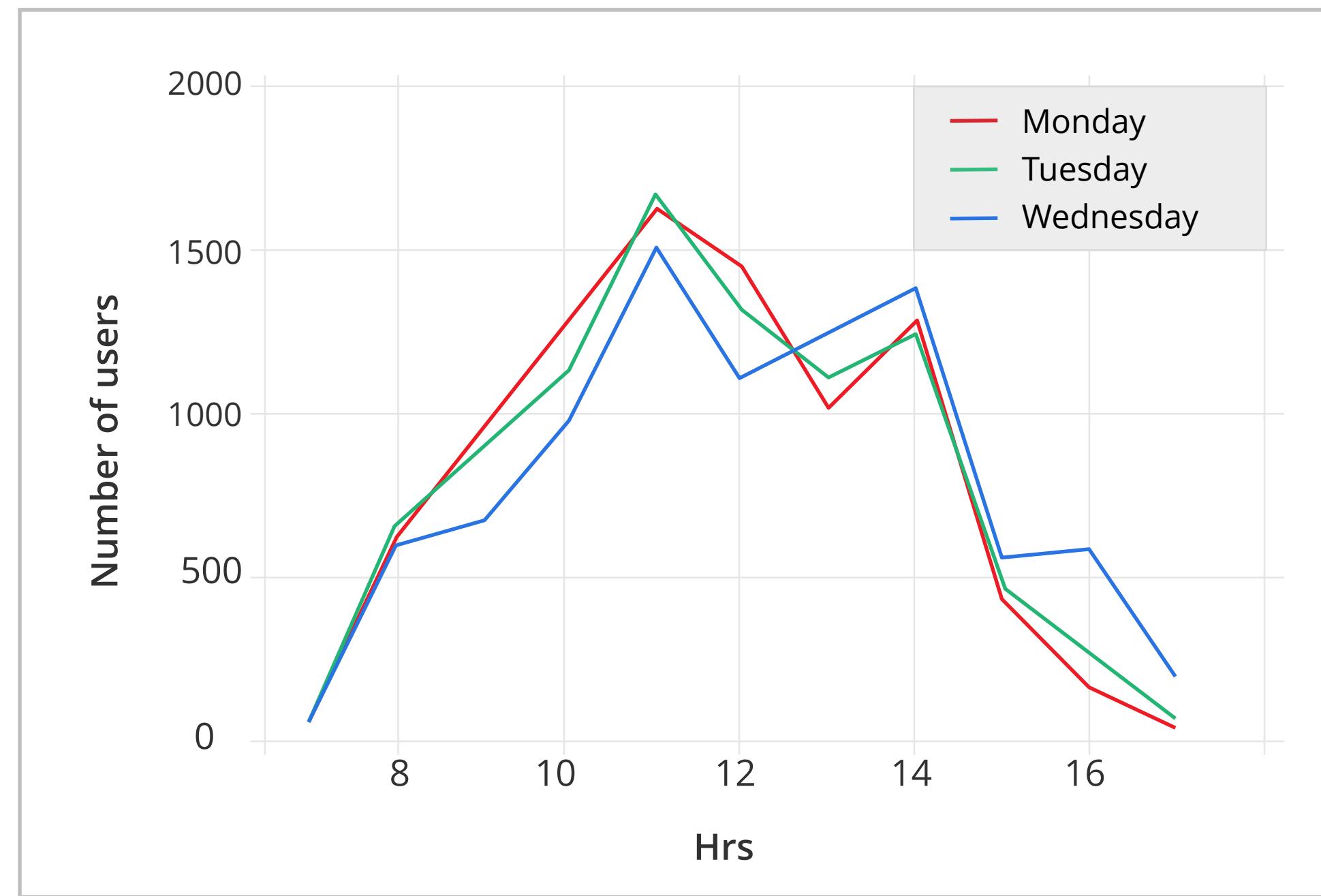
Web traffic data

```
In [5]: #select the style of the plot  
style.use('ggplot')  
#plot the web site traffic data (X-axis hrs and Y axis as number of users)  
#plot the monday web traffic with red color  
plt.plot(time_hrs,web_monday,'r',label='monday',linewidth=1)  
#plot the monday web traffic with green color  
plt.plot(time_hrs,web_tuesday,'g',label='tuesday',linewidth=1.5)  
#plot the monday web traffic with blue color  
plt.plot(time_hrs,web_wednesday,'b',label='wednesday',linewidth=2)  
plt.axis([6.5,17.5,50,2000])  
#set the title of the plot  
plt.title('Web site traffic')  
#set label for x axis  
plt.xlabel('Hrs')  
#set label for y axis  
plt.ylabel('Number of users')  
plt.legend()  
plt.show()
```

Set different colors and line widths for different days

# Multiple Plots

## Web site traffic



# Subplots

Subplots are used to display multiple plots in the same window.

With subplot, you can arrange plots in a regular grid.

The syntax for subplot is

`subplot(m,n,p).`



It divides the current window into an m-by-n grid and creates an axis for a subplot in the position specified by p.

For example,

`subplot(2,1,2)` creates two subplots which are stacked vertically on a grid.

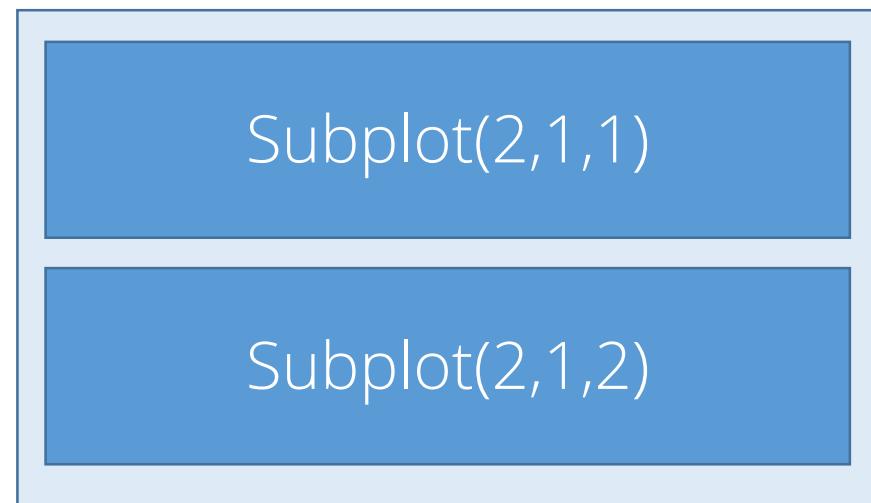
`subplot(2,1,4)` creates four subplots in one window.

## Subplots (contd.)

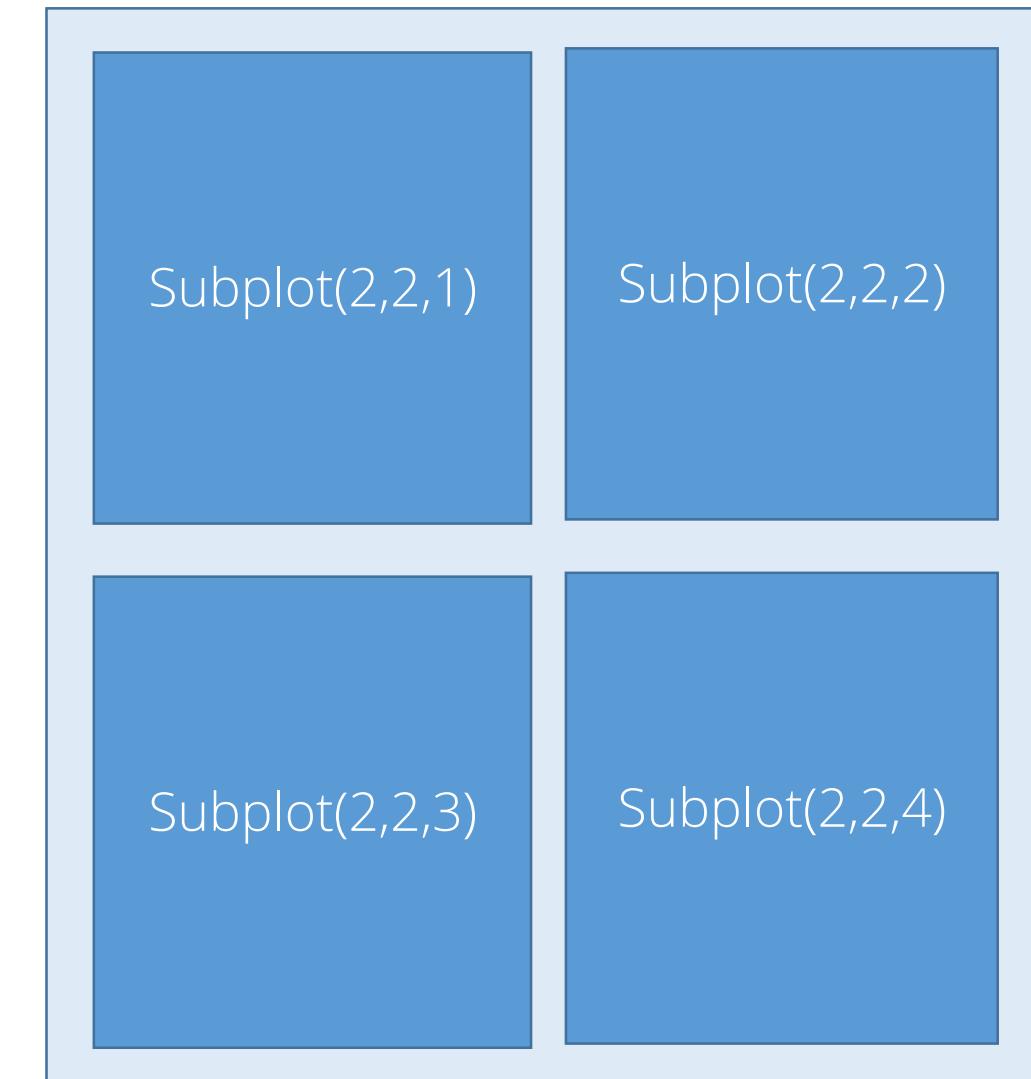
Subplots are used to display multiple plots in the same window.

With subplots, you can arrange plots in a regular grid.

Grid divided  
into two  
vertically  
stacked plots



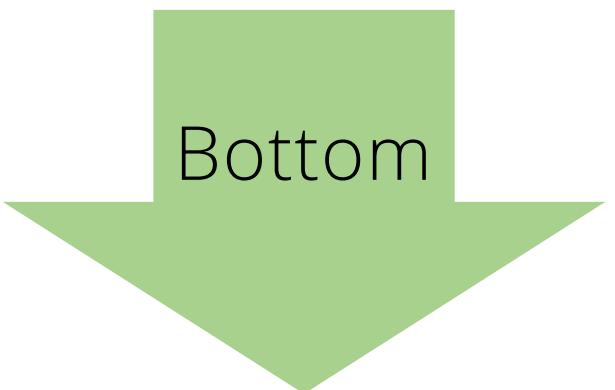
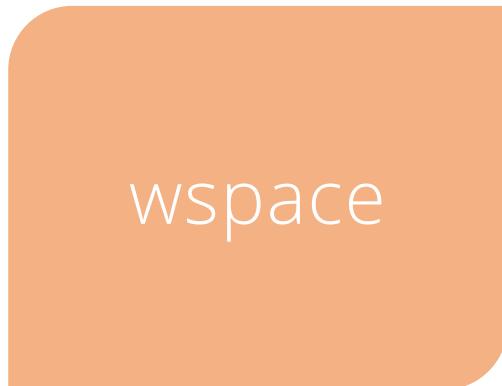
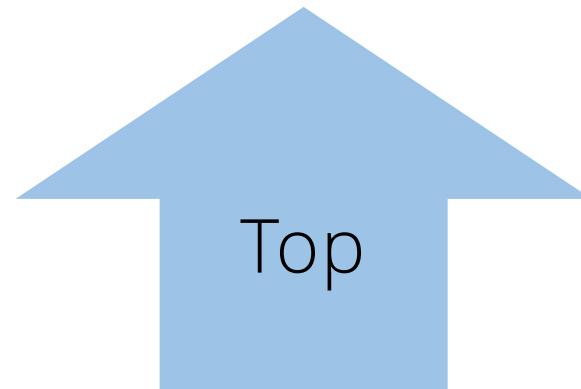
Grid divided  
into four plots



# Layout

Layout and Spacing adjustments are two important factors to be considered while creating subplots.

Use the plt.subplots\_adjust() method with the parameters hspace and wspace to adjust the distances between the subplots and move them around on the grid.





# Knowledge Check

KNOWLEDGE  
CHECK

**Which of the following methods is used to adjust the distances between the subplots?**

- a. plot.subplots\_adjust()
- b. plt.subplots\_adjust()
- c. subplots\_adjust()
- d. plt.subplots.adjust()



KNOWLEDGE  
CHECK

**Which of the following methods is used to adjust the distances between the subplots?**

- a. plot.subplots\_adjust()
- b. plt.subplots\_adjust()
- c. subplots\_adjust()
- d. plt.subplots.adjust()



The correct answer is . b.

Explanation plt.subplots\_adjust() used to adjust the distances between the subplot**s**.

# Types of Plots

---

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

Scatter Plot

Heat Map

Pie Chart

Error Bar

## Types of Plots (contd.)

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

Histograms are graphical representations of a probability distribution. A histogram is a kind of a bar chart.

Scatter Plot

Using matplotlib and its bar chart function, you can create histogram charts.

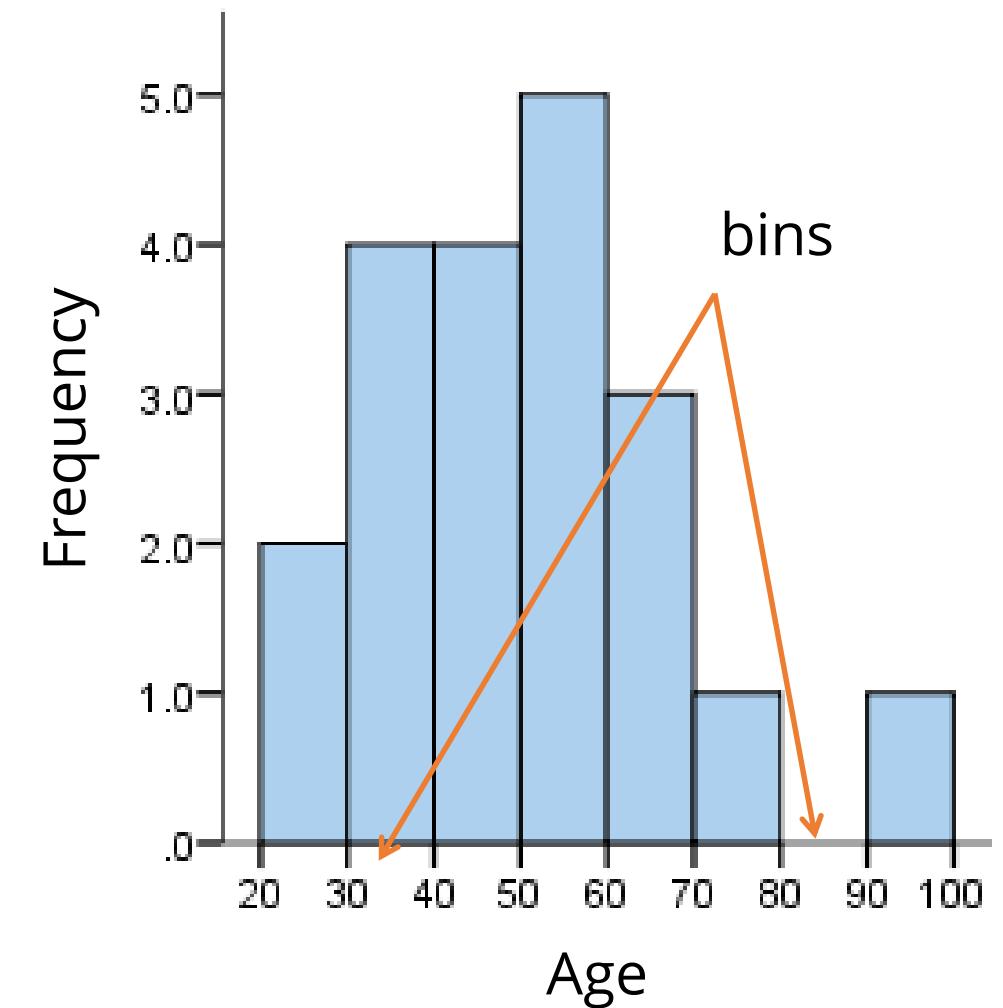
Heat Map

Advantages of Histogram charts:

Pie Chart

- They display the number of values within a specified interval.
- They are suitable for large datasets as they can be grouped within the intervals.

Error Bar



## Types of Plots (contd.)

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

A scatter plot is used to graphically display the relationships between variables.

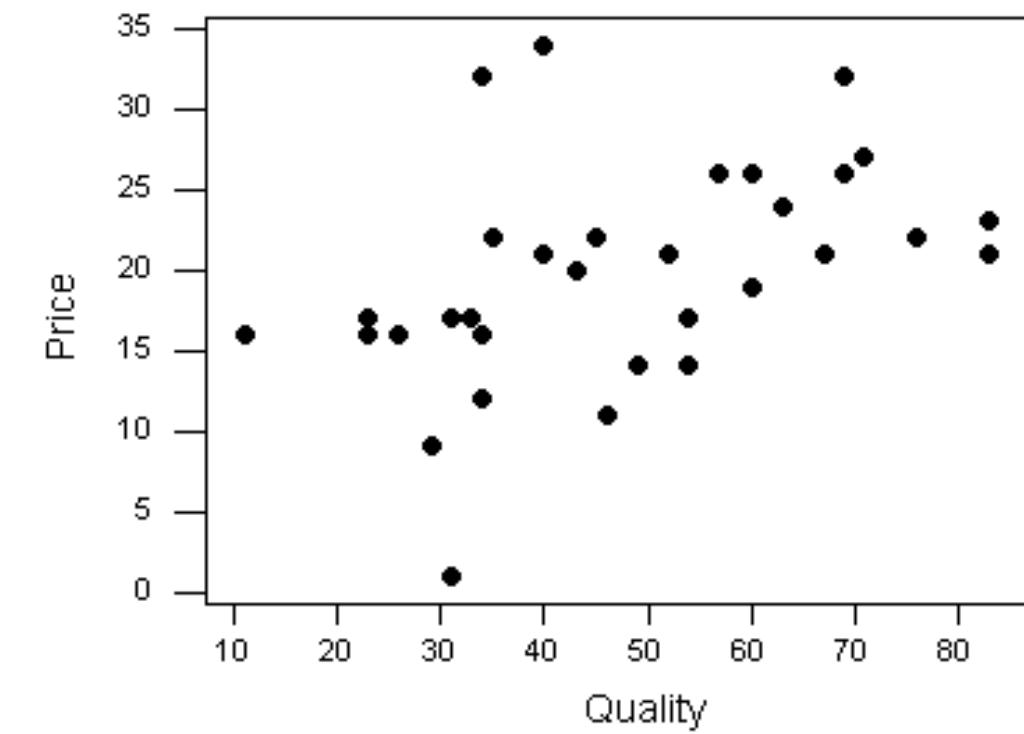
Scatter Plot

It has several advantages:

- Shows the correlation between variables
- Is suitable for large datasets
- Is easy to find clusters
- Is possible to represent each piece of data as a point on the plot

Pie Chart

Error Bar



## Types of Plots (contd.)

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

A heat map is a way to visualize two-dimensional data. Using heat maps, you can gain deeper and faster insights about data than other types of plots.

Scatter Plot

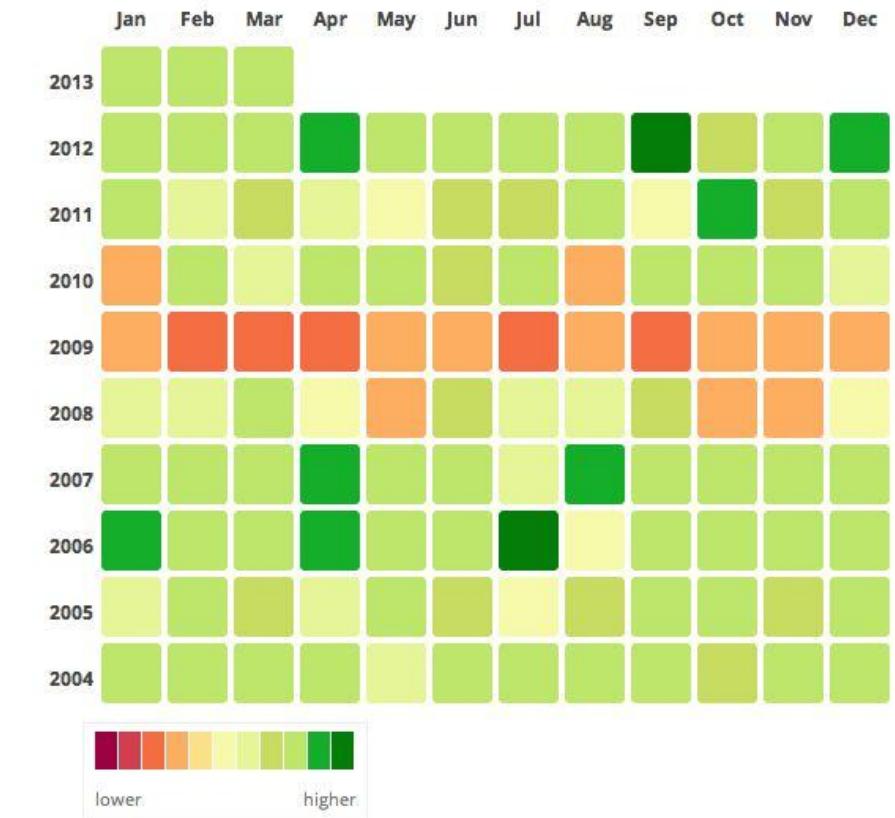
It has several advantages:

- Draws attention to the risk-prone area
- Uses the entire dataset to draw meaningful insights
- Is used for cluster analysis and can deal with large datasets

Pie Chart

Error Bar

Monthly change from previous year



## Types of Plots (contd.)

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

Scatter Plot

Heat Map

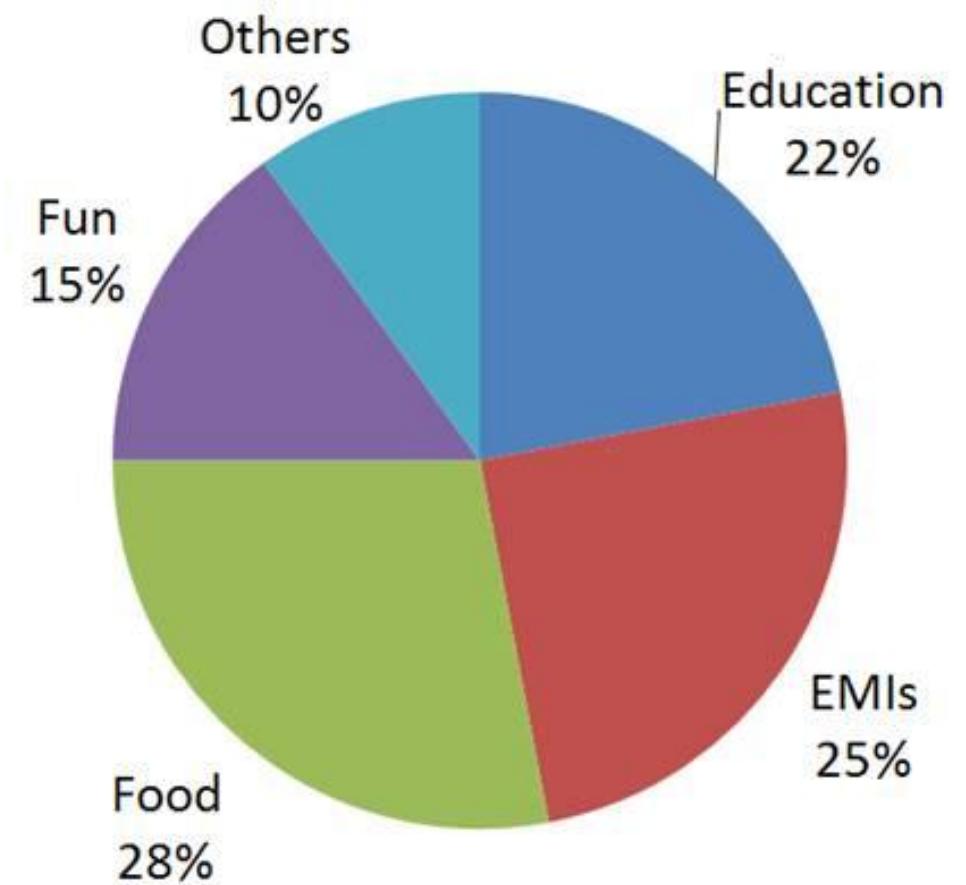
Pie Chart

Error Bar

Pie charts are used to show percentage or proportional data. matplotlib provides the pie() method to create pie charts.

It has several advantages:

- Summarizes a large dataset in visual form
- Displays the relative proportions of multiple classes of data
- Size of the circle is made proportional to the total quantity



## Types of Plots (contd.)

You can create different types of plots using matplotlib:

*Click each plot to know more.*

Histogram

An error bar is used to graphically represent the variability of data. It is used mainly to identify errors. It builds confidence about the data analysis by revealing the statistical difference between the two groups of data.

Scatter Plot

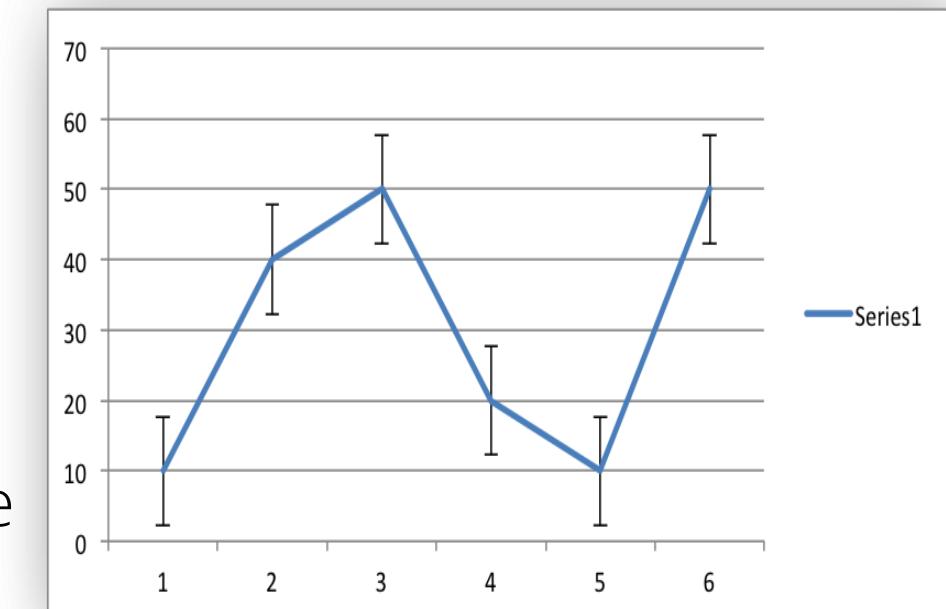
Heat Map

Pie Chart

Error Bar

It has several advantages:

- Shows the variability in data and indicates the errors.
- Depicts the precision in the data analysis.
- Demonstrates how well a function and model are used in the data analysis.
- Describes the underlying data.



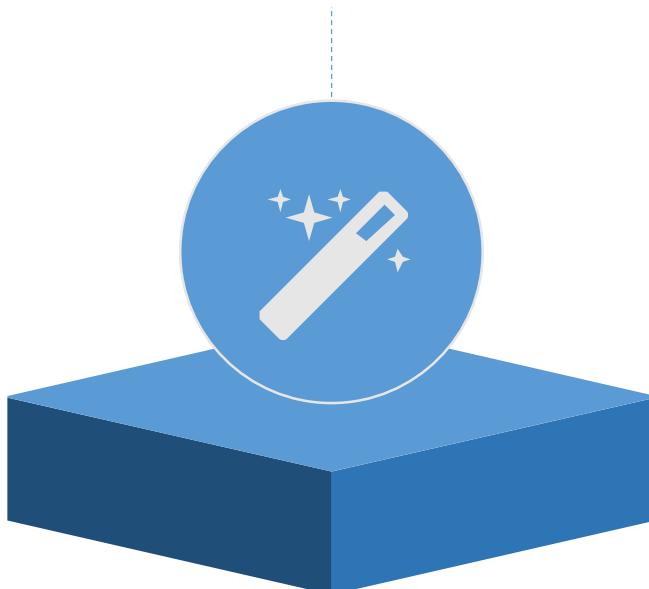
# Seaborn

---

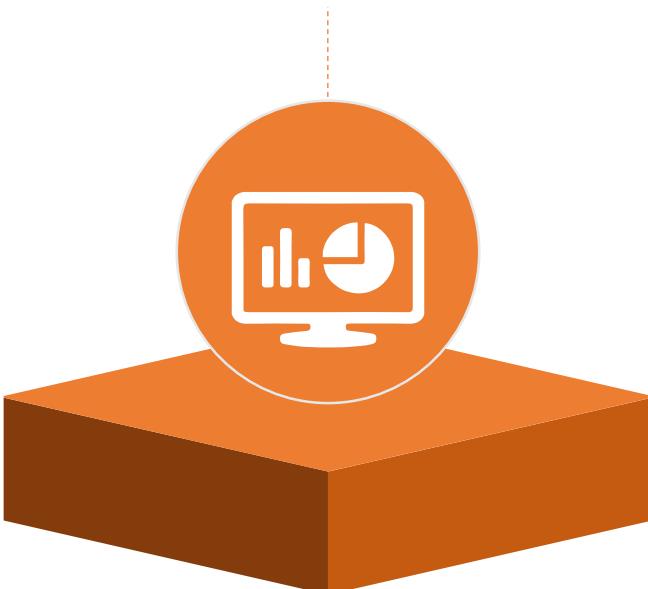
Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface to draw attractive statistical graphics.

There are several advantages:

Possesses built-in themes for better visualizations



Has built-in statistical functions which reveal hidden patterns in the dataset



Has functions to visualize matrices of data





Problem

Instructions

Analyze the “auto mpg data” and draw a pair plot using seaborn library for mpg, weight, and origin.

Sources:

(a) Origin: This dataset was taken from the StatLib library maintained at Carnegie Mellon University.

- Number of Instances: 398
- Number of Attributes: 9 including the class attribute
- Attribute Information:
  - mpg: continuous
  - cylinders: multi-valued discrete
  - displacement: continuous
  - horsepower: continuous

Problem

Instructions

- weight: continuous
- acceleration: continuous
- model year: multi-valued discrete
- origin: multi-valued discrete
- car name: string (unique for each instance)



Problem

Instructions

You have been provided with a dataset that lists Ohio State's leading causes of death from the year 2012.

Using the two data points:

- Cause of deaths and
- Percentile

Draw a pie chart to visualize the dataset.

Problem

Instructions

Instructions to perform the assignment:

- Download the dataset “Ohio\_State\_data”. Use the data provided to create relevant and required variables.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



**QUIZ**  
**1**

**Which of the following libraries needs to be imported to display the plot on Jupyter notebook?**

- a. %matplotlib
- b. %matplotlib inline
- c. import matplotlib
- d. import style



**QUIZ**  
**1**

**Which of the following libraries needs to be imported to display the plot on Jupyter notebook?**

- a. %matplotlib
- b. %matplotlib inline
- c. import matplotlib
- d. import style



The correct answer is **b**.

**Explanation:** To display the plot on Jupyter notebook “import%matplotlib inline.”

**QUIZ**  
**2**

**Which of the following keywords is used to decide the transparency of the plot line?**

- a. Legend
- b. Alpha
- c. Animated
- d. Annotation



**QUIZ**  
**2**

**Which of the following keywords is used to decide the transparency of the plot line?**

- a. Legend
- b. Alpha
- c. Animated
- d. Annotation



The correct answer is **C.**

**Explanation:** Alpha decides the line transparency in line properties while plotting line plot/ chart.

**QUIZ**  
**3**

**Which of the following plots is used to represent data in a two-dimensional manner?**

- a. Histogram
- b. Heat Map
- c. Pie Chart
- d. Scatter Plot



**QUIZ****3**

**Which of the following plots is used to represent data in a two-dimensional manner?**

- a. Histogram
- b. Heat Map
- c. Pie Chart
- d. Scatter Plot



The correct answer is **b**.

**Explanation:** Heat Maps are used to represent data in a two-dimensional manner.

**QUIZ****4**

**Which of the following statements limits both x and y axes to the interval [0, 6]?**

- a. plt.xlim(0, 6)
- b. plt.ylim(0, 6)
- c. plt.xylim(0, 6)
- d. plt.axis([0, 6, 0, 6])



**QUIZ****4**

**Which of the following statements limits both x and y axes to the interval [0, 6]?**

- a. plt.xlim(0, 6)
- b. plt.ylim(0, 6)
- c. plt.xylim(0, 6)
- d. plt.axis([0, 6, 0, 6])

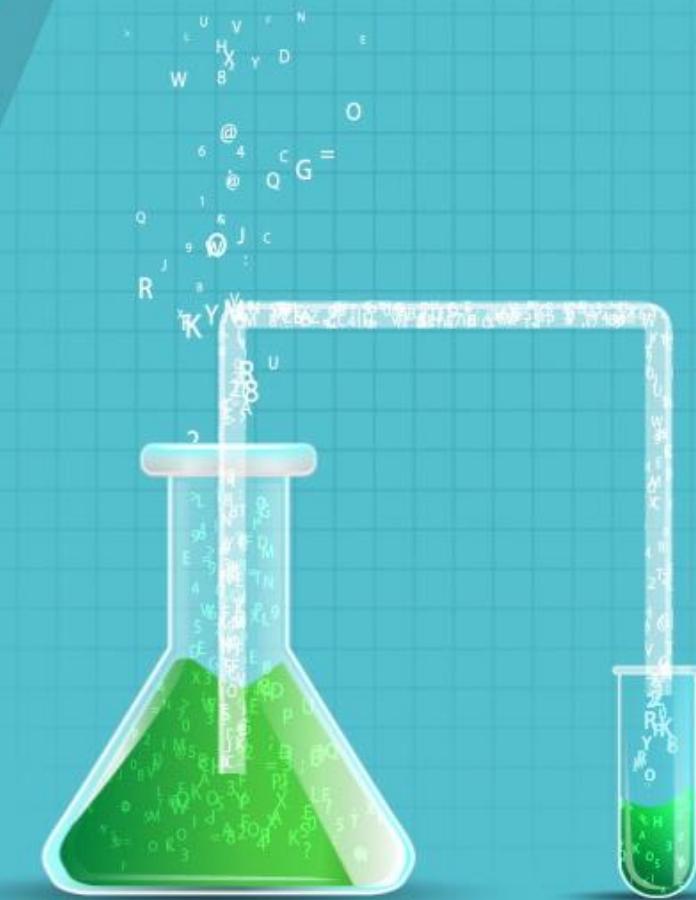


The correct answer is **d**.

**Explanation:** plt.axis([0, 6, 0, 6]) statement limits both x and y axes to the interval [0, 6].

# Key Takeaways

- Data visualization is the technique to present the data in a pictorial or graphical format.
- There are three major considerations for data visualization. They are clarity, accuracy, and efficiency.
- The matplotlib is a python 2D plotting library for data visualization and the creation of interactive graphics/ plots.
- A plot is a graphical representation of data which shows the relationship between two variables or the distribution of data.
- Subplots are used to display multiple plots in the same window.
- Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface to draw attractive statistical graphics.



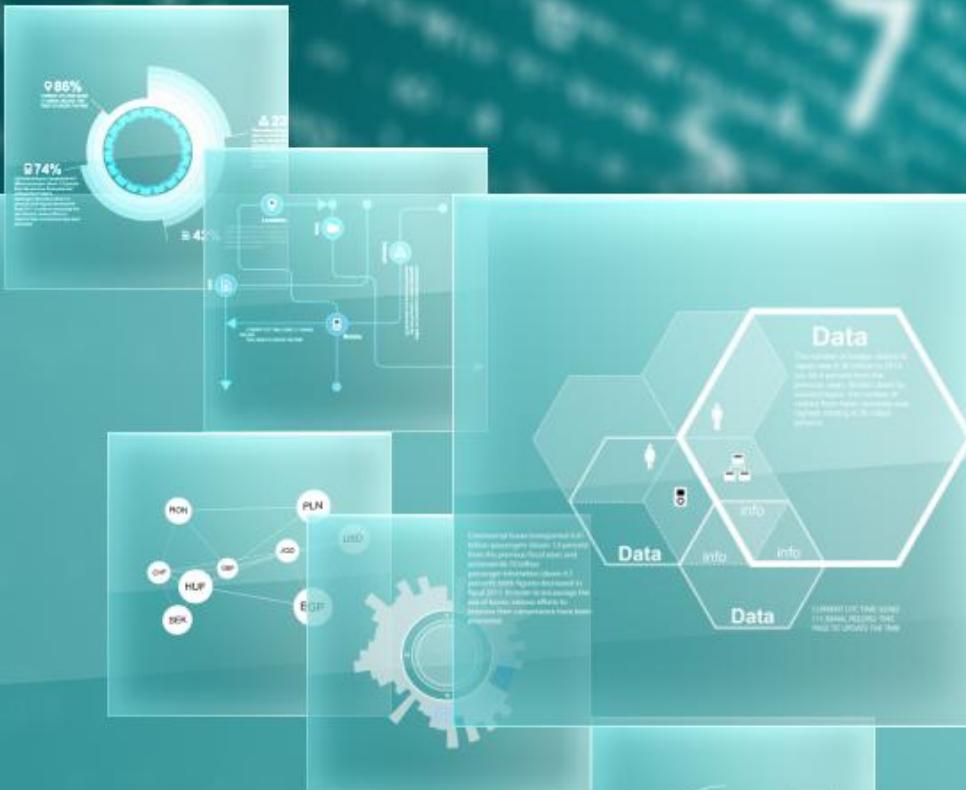
**This concludes “Data Visualization in Python using matplotlib”**

The next lesson is “Web Scraping with BeautifulSoup.”

DATA  
SCIENCE

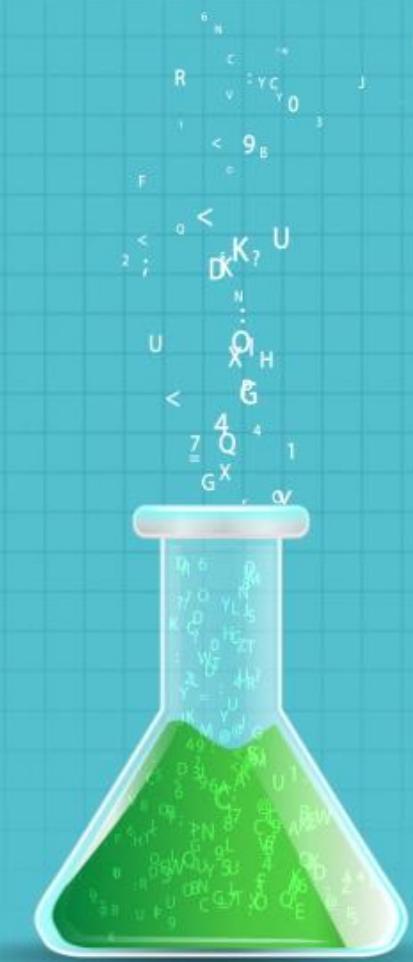
# Data Science with Python

## Lesson 11 — Web Scraping with BeautifulSoup



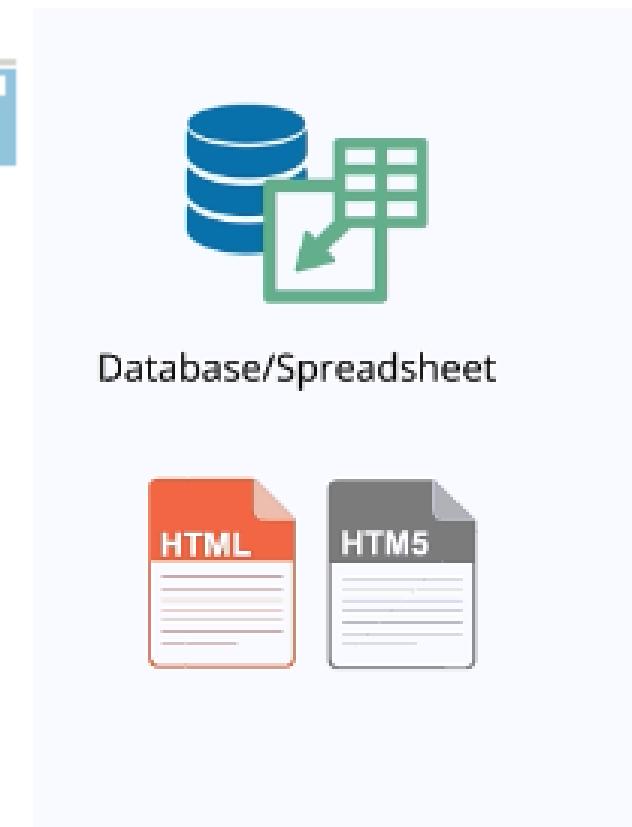
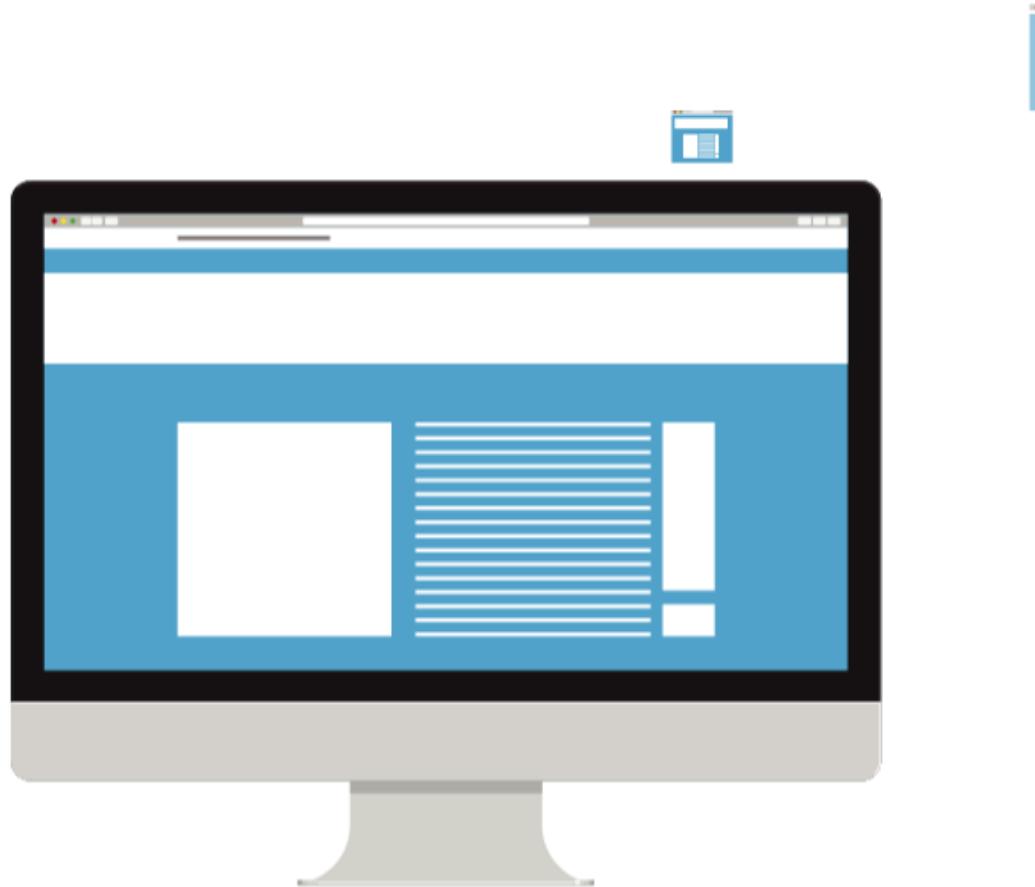
# What's In It For Me

- Define web scraping and explain the importance of web scraping
- Lists the steps involved in the web scraping process
- Describe basic terminologies such as parser, object, and tree associated with the BeautifulSoup
- Understand various operations such as searching, modifying, and navigating the tree to yield the required result



# What is Web Scraping

Web scraping is a computer software technique of extracting information from websites in an automated fashion.



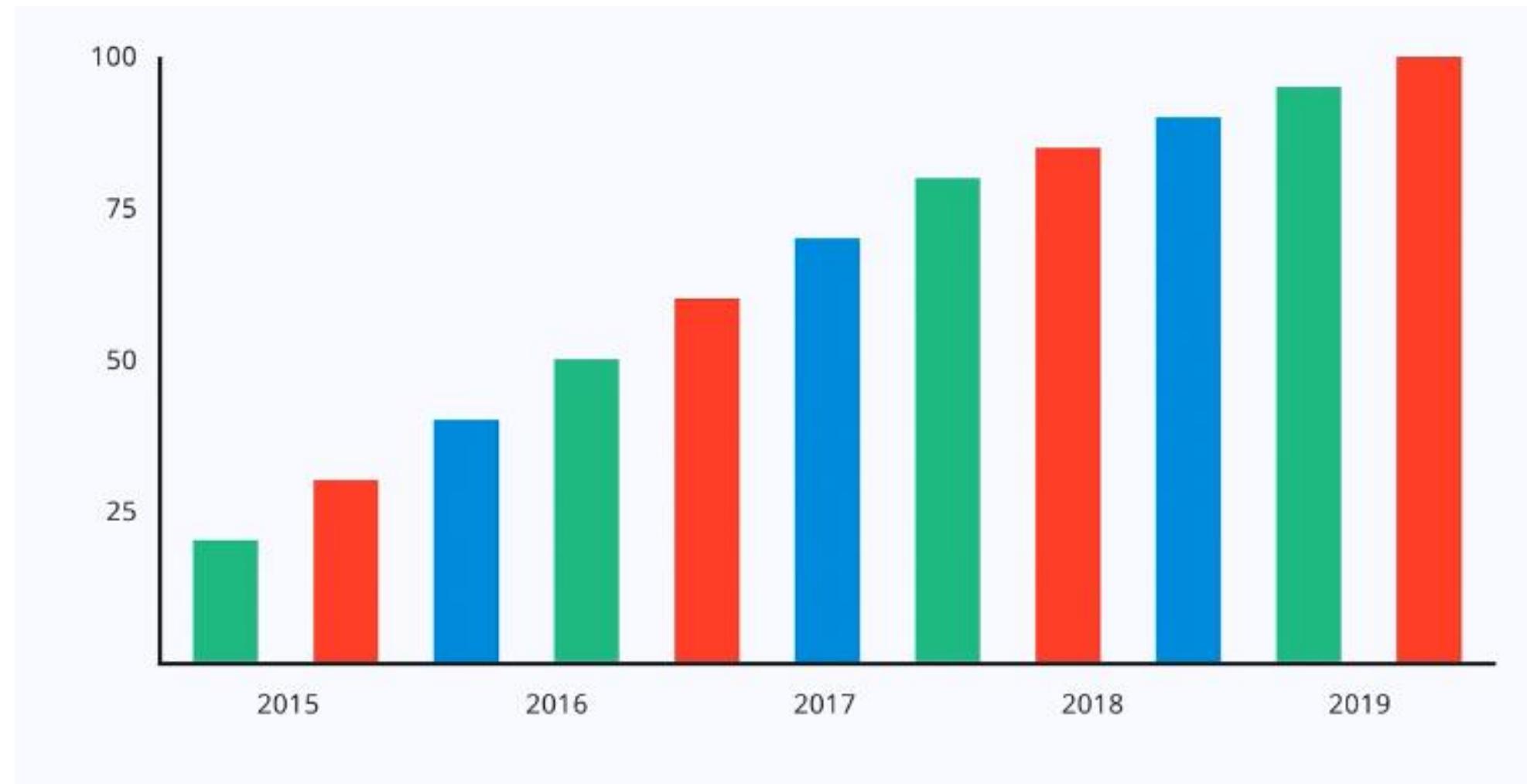
# What Web Scraping is (contd.)

Web scraping is a computer software technique of extracting information from websites in an automated fashion.



# Why Web Scraping

Every day, you find yourself in a situation where you need to extract data from the web.



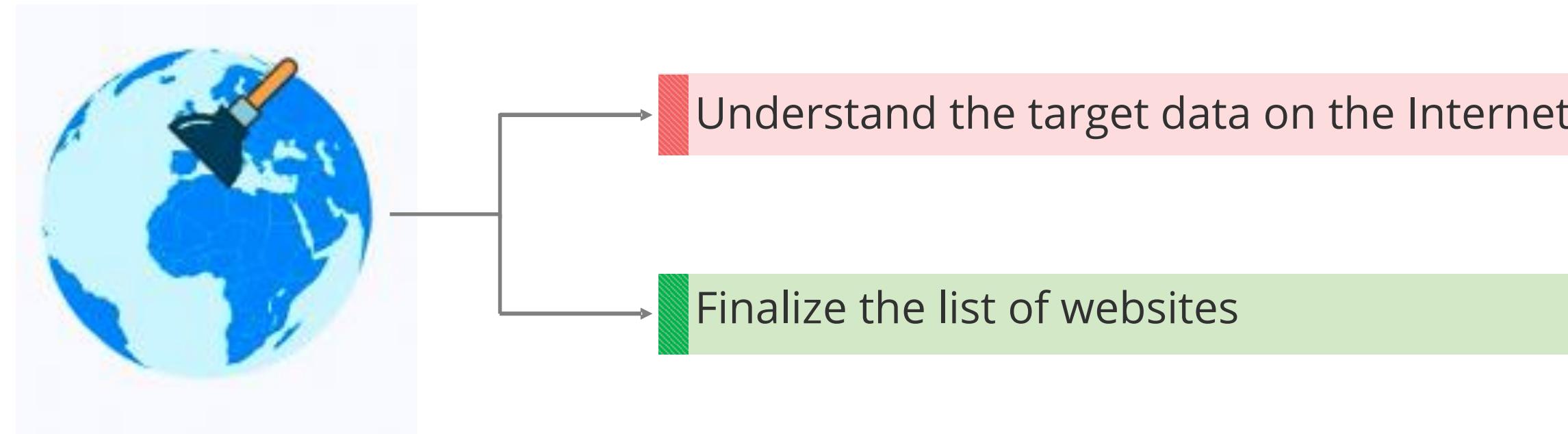
## **Why Web Scraping (contd.)**

Every day, you find yourself in a situation where you need to extract data from the web.



# Web Scraping Process—Basic Preparation

There are two basic things to consider before setting up the web scraping process.



## **Web Scraping Process (contd.)**

Once you have understood the target data and finalized the list of websites, you need to design the web scraping process.

The steps involved in a typical web scraping process are as follows:



Step 1: A web request is sent to the targeted website to collect the required data.

## **Web Scraping Process (contd.)**

Once you have understood the target data and finalized the list of websites, you need to design the web scraping process.

The steps involved in a typical web scraping process are as follows:



Step 2: The information is retrieved from the targeted website in HTML or XML format from web.

## Web Scraping Process (contd.)

Once you have understood the target data and finalized the list of websites, you need to design the web scraping process.

The steps involved in a typical web scraping process are as follows:

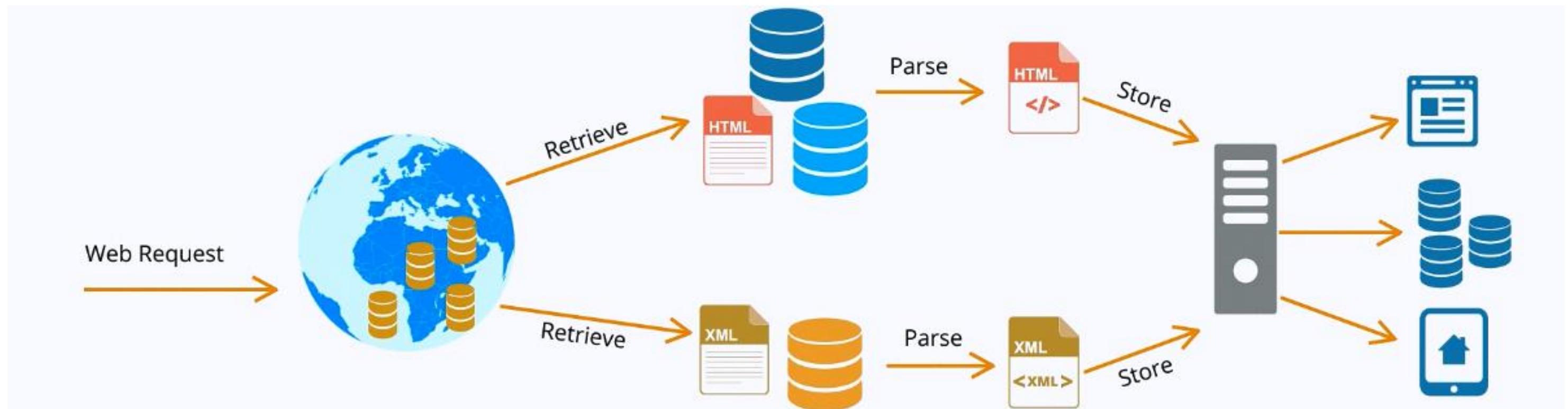


Step 3: The retrieved information is parsed to the several parsers based on the data format. Parsing is a technique to read data and extract information from the available document.

## Web Scraping Process (contd.)

Once you have understood the target data and finalized the list of websites, you need to design the web scraping process.

The steps involved in a typical web scraping process are as follows:



Step 4: The parsed data is stored in the desired format. You can follow the same process to scrap another targeted web.

# Web Scraping Software vs. Web Browser

A web scraping software will interact with websites in the same way as your web browser.

A Web scraper is used to extract the information from web in routine and automated manner.

## Web Browser

The screenshot shows a web browser window with the URL [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser). The page content is about web browsers, featuring a graph titled 'Usage share of web browsers according to StatCounter' showing trends from 1995 to 2015. Below the graph is a network visualization titled 'Internet' showing routing paths. The browser interface includes a sidebar with navigation links like 'History' and 'Marc Andreessen, inventor'.

Displays the data

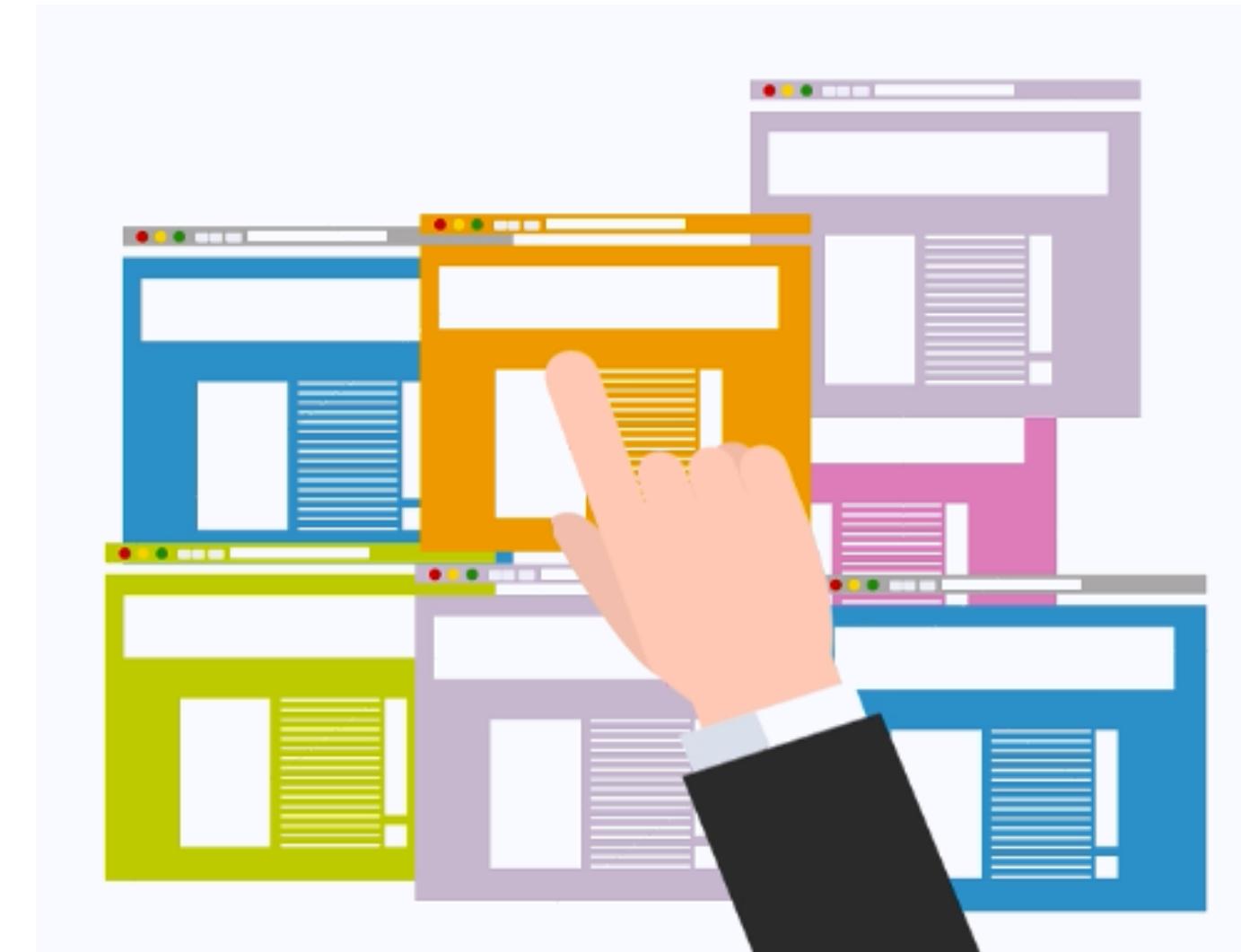
## Web Scraping Software

The screenshot shows a software interface for web scraping. On the left, a tree view labeled 'My Agents' lists categories: Demo, Inputs, Output, Production, and Unit\_Testing. Under 'Unit\_Testing', there are numerous sub-tasks such as 'HTML\_Table\_extraction.scraping', 'Links\_extraction.scraping', etc. To the right, a table lists 'HYPER\_LINK' and 'ANCHOR\_TEXT' for various URLs, with the first row highlighted in blue. At the bottom, a log window shows the execution of 'Links\_extraction.scraping' on the URL [http://cdn.datascraping.co/sample\\_content/links.html](http://cdn.datascraping.co/sample_content/links.html), detailing steps like 'Executing', 'Requesting', 'Loaded', 'Traversed', and 'Extraction Completed'.

Saves data from the web page to the local file or database

# Web Scraping Considerations

It's important to read and understand the legal information and terms and conditions mentioned in the website.

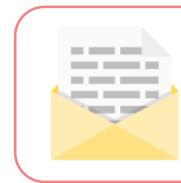


# Web Scraping Considerations (contd.)

It's important to read and understand the legal information and terms and conditions mentioned in the website.



Legal Constraints



Notice



Copyright



Trademark Material



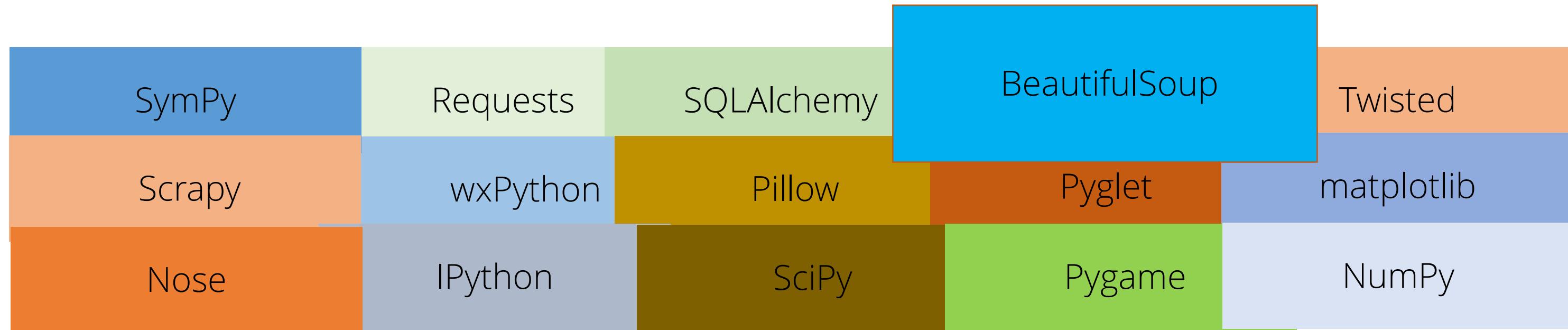
Patented Information

# Web Scraping Tool—BeautifulSoup



# Web Scraping Tool: BeautifulSoup (contd.)

BeautifulSoup, is an easy, intuitive, and a robust Python library designed for web scraping.



# Web Scraping Tool: BeautifulSoup (contd.)

BeautifulSoup, is an easy, intuitive, and a robust Python library designed for web scraping.

Some of the reasons to choose BeautifulSoup are as follows:



Efficient tool for dissecting documents and extracting information from the web pages



Powerful sets of built-in methods for navigating, searching, and modifying a parse tree



Possess parser that supports both html and xml documents

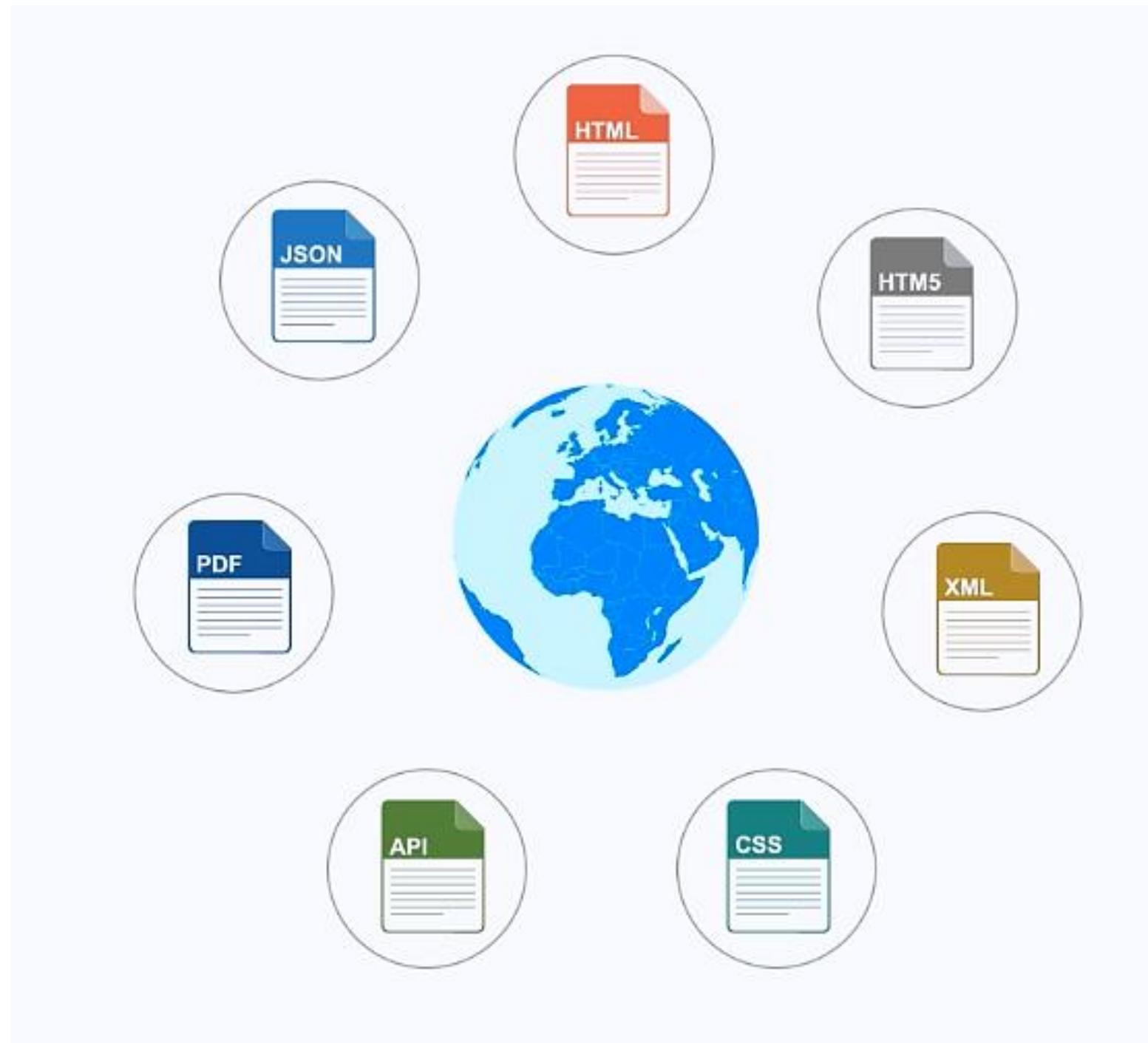


Converts all incoming documents to Unicode automatically

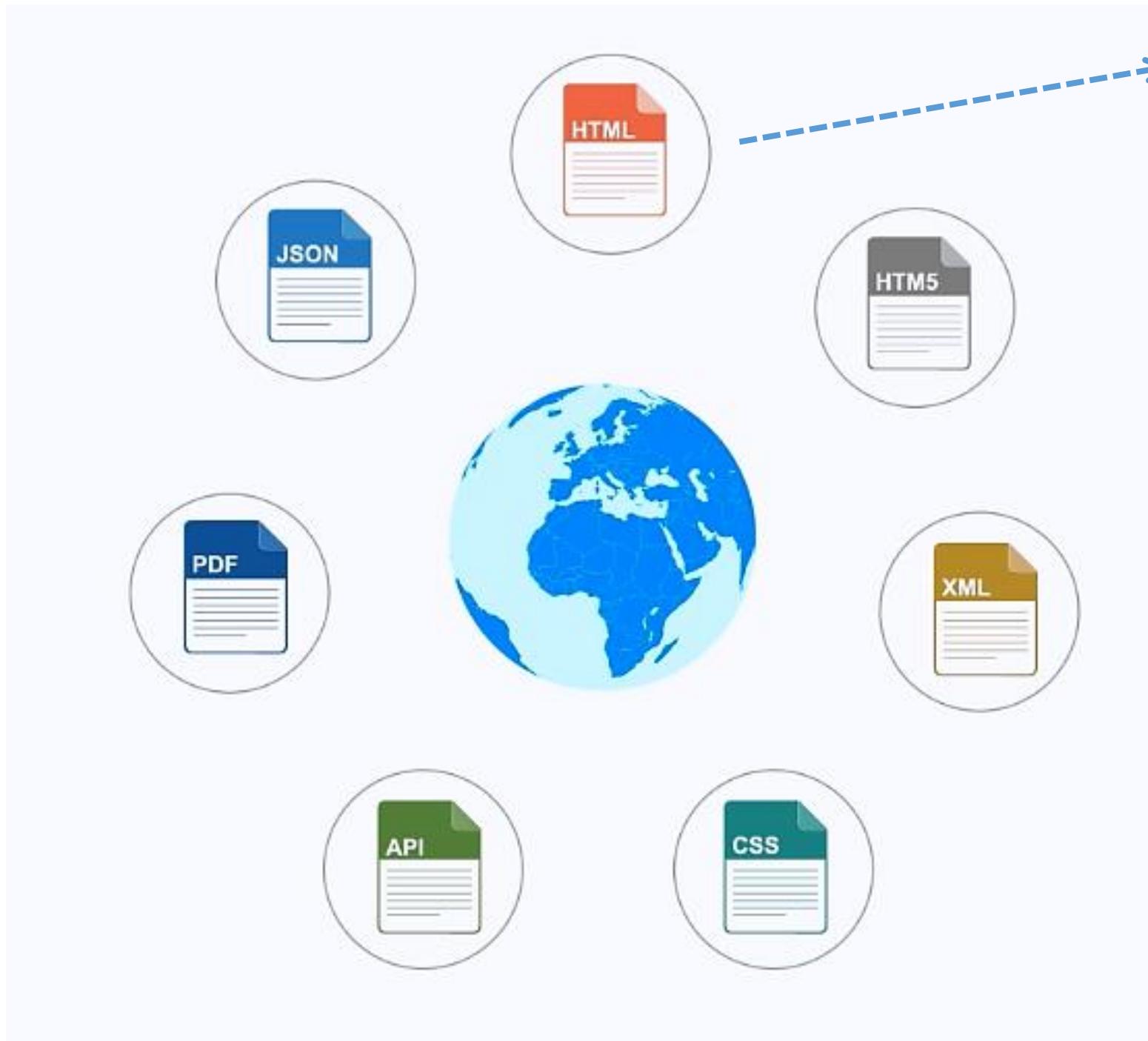


Converts all outgoing documents to UTF-8 automatically

# Common Data/ Page Formats on The Web

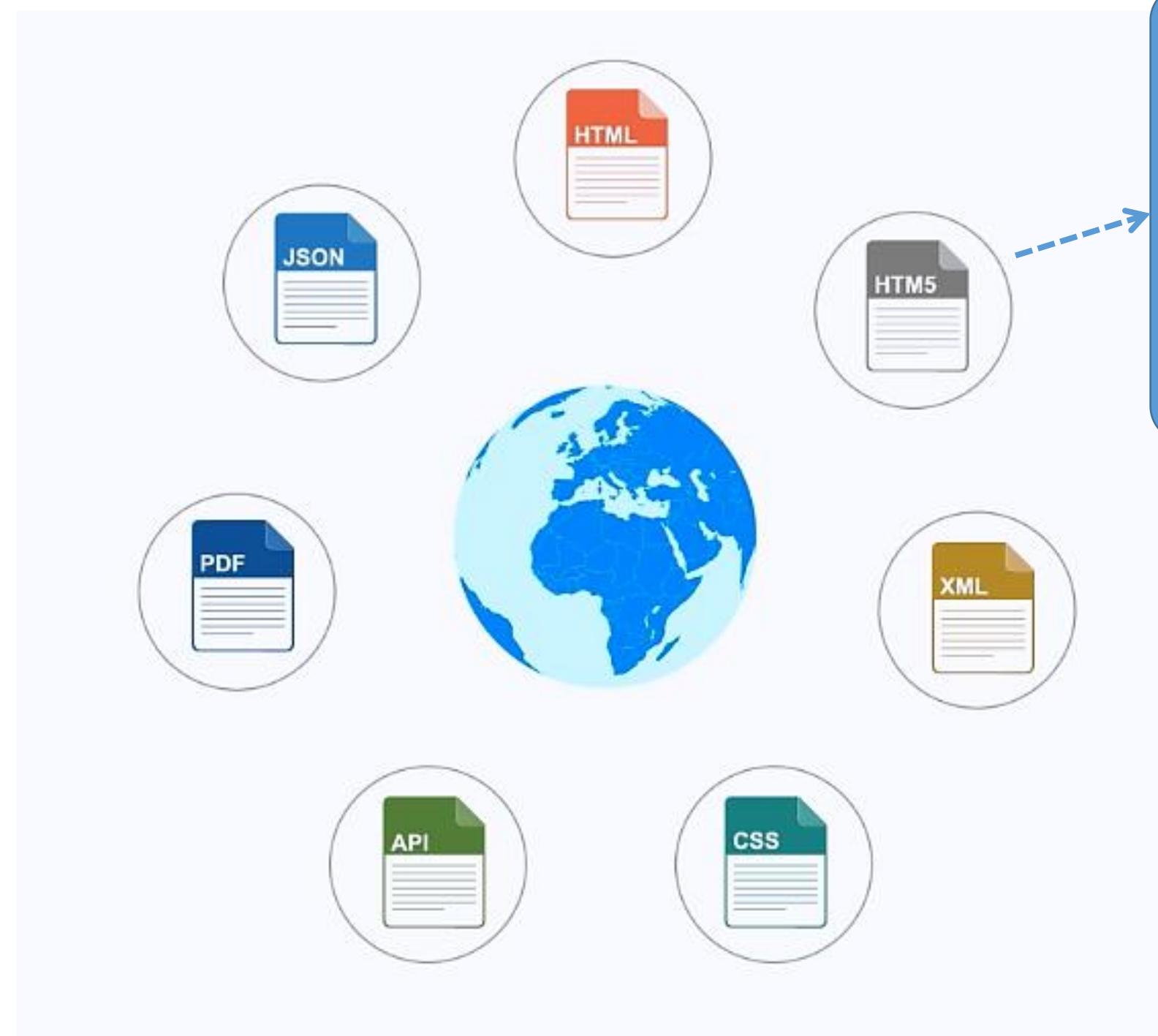


# Common Data/ Page Formats on The Web (contd.)



An HTML page is one of the oldest, easiest, and the most popular methods to upload information on the web.

## Common Data/ Page Formats on The Web (contd.)



An HTML 5 is a new HTML standard which gained popularity with the mobile devices.

# Common Data/ Page Formats on The Web (contd.)



# Common Data/ Page Formats on The Web (contd.)



# Common Data/ Page Formats on The Web (contd.)



Application Program Interface, or APIs, has now become a common practice to extract information from the web.

# Common Data/ Page Formats on The Web (contd.)

PDF is also widely used to upload information and reports.



# Common Data/ Page Formats on The Web (contd.)

JavaScript Object Notation, or JSON, is a lightweight and popular format used for information exchange on the web.



# The Parser

---



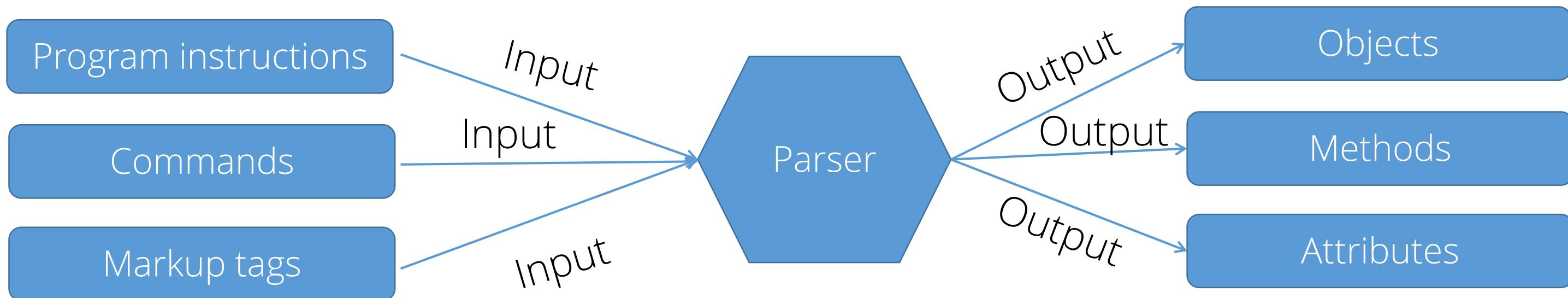
What is a parser?

How does it help Data Scientists in  
the web scraping process?

# The Parser

A Parser is a basic tool to interpret or render information from a web document.

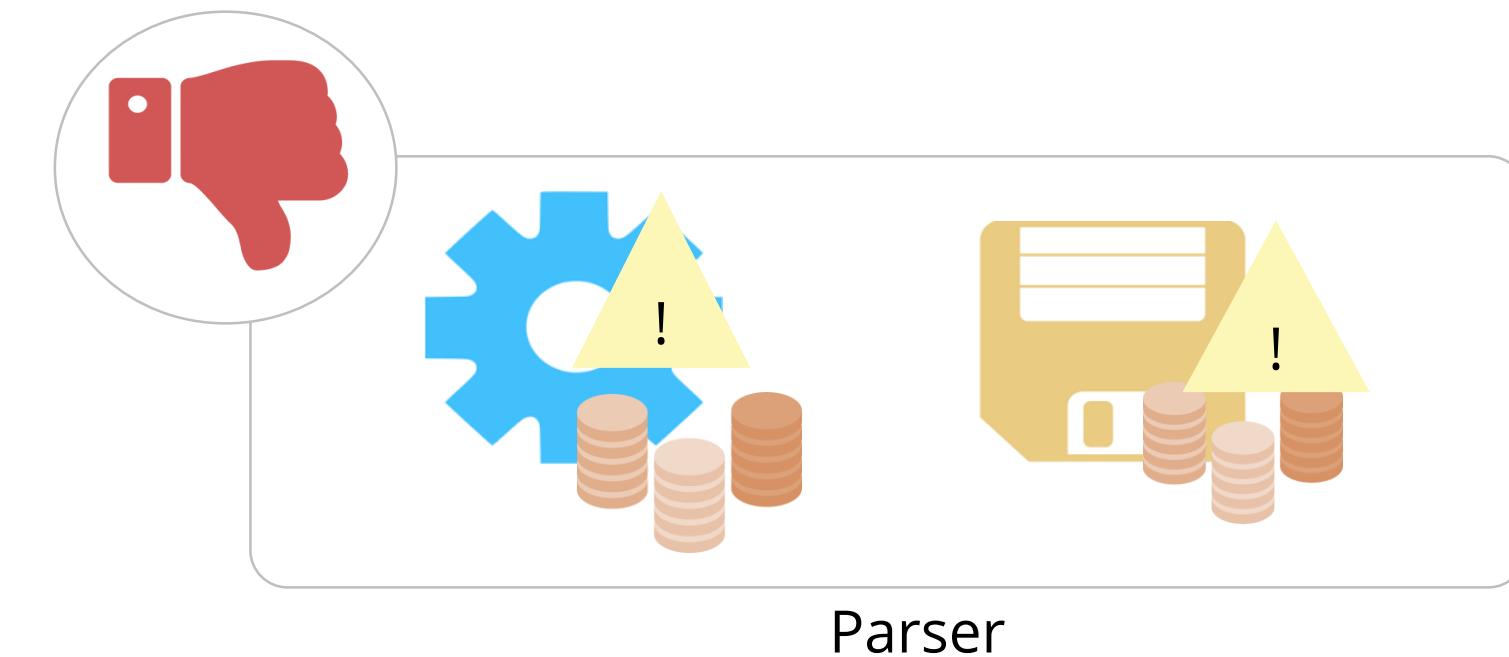
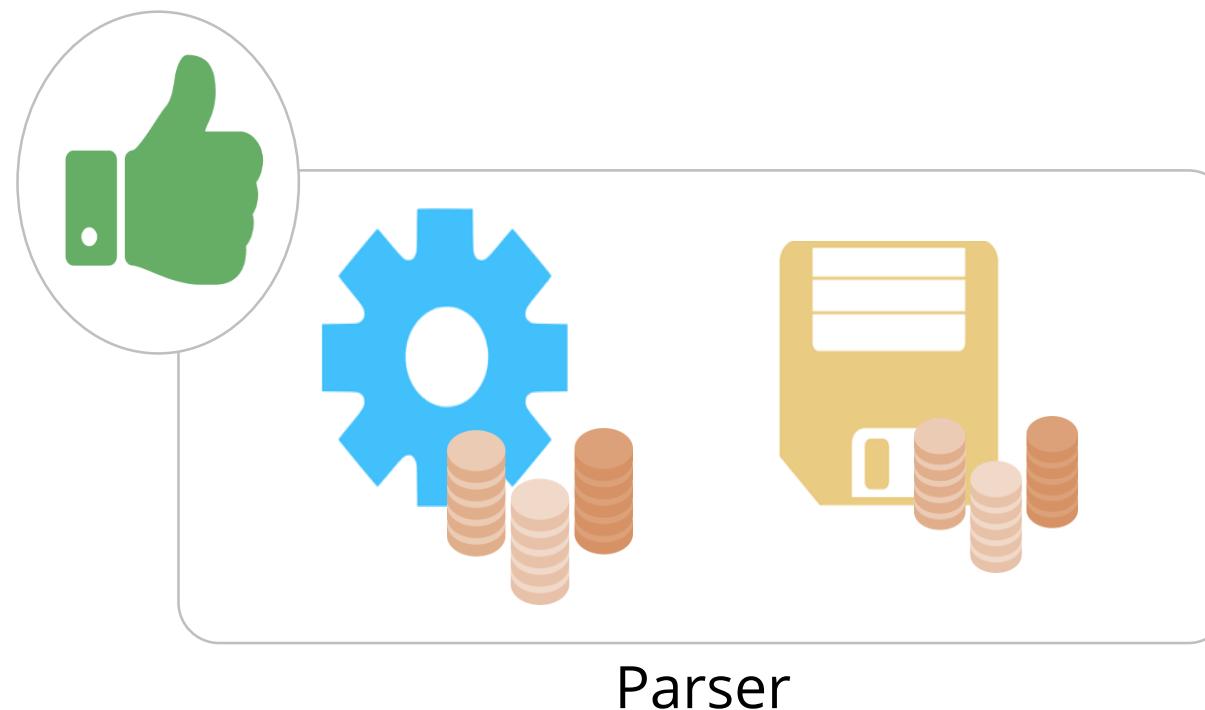
A Parser is also used to validate the input information before processing it.



# Importance of Parsing

Parsing data is one of the most important steps in the web scraping process.

Failing to parse the data would eventually lead to a failure of the entire process.



# Various Parser

There are various parsers supported by BeautifulSoup:

html.parser

HTML parser is Python based, fast, and lenient.

lxml html

Lxml html is not built using Python and it depends on C. However, it is fast and lenient in nature.

lxml xml

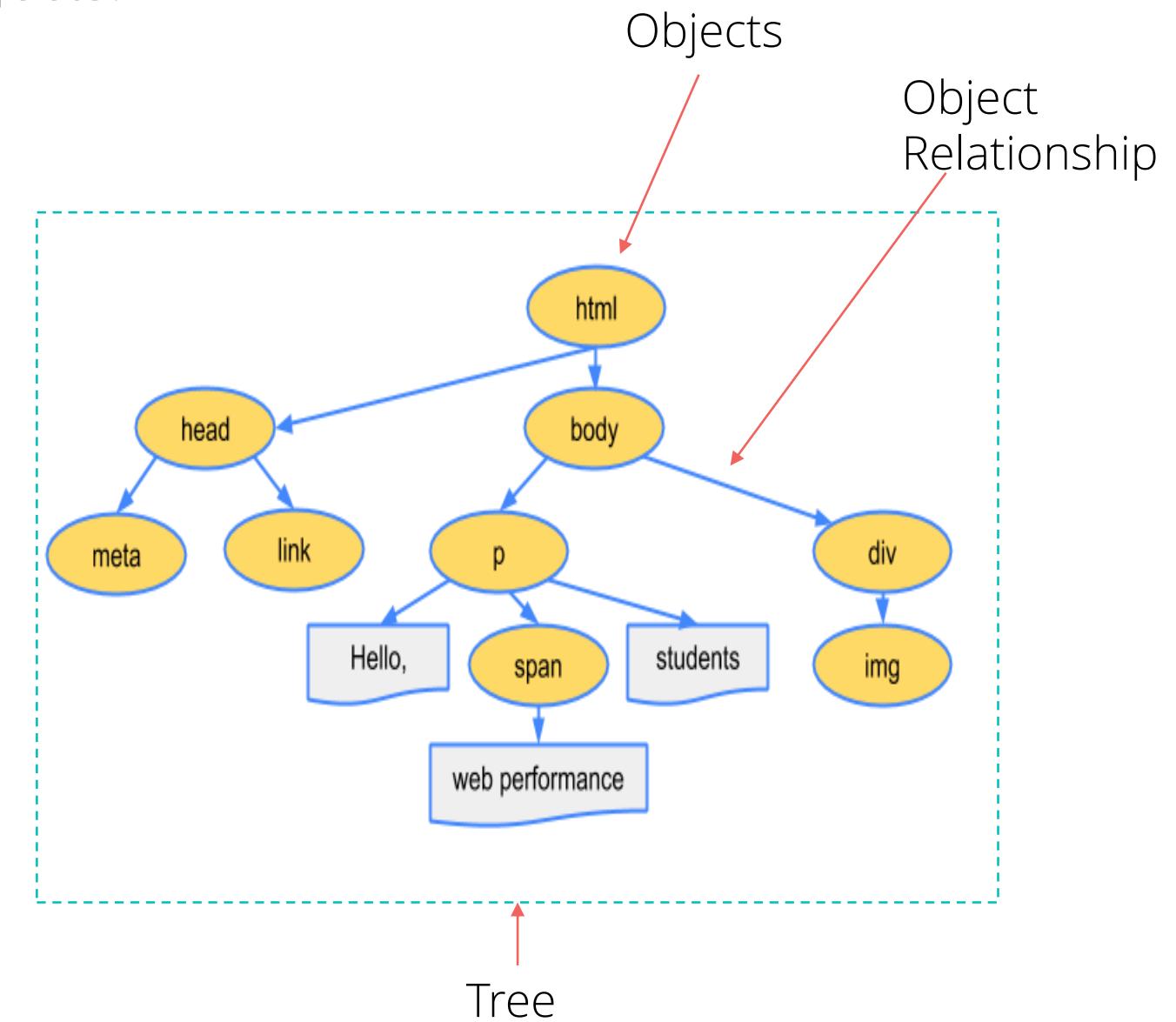
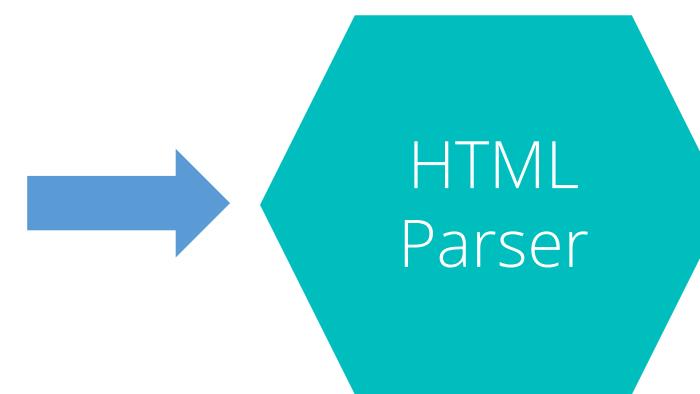
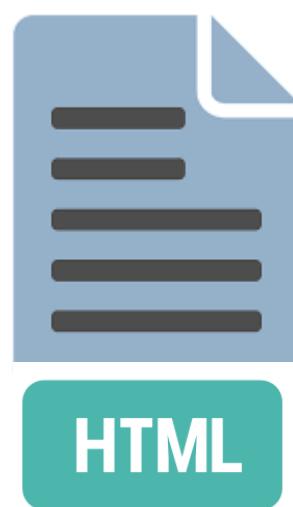
Lxml xml is the only xml parser available and it also depends on C.

html5lib

HTML 5lib is another Python-based parser; however, it is slow and able to create valid HTML5.

# Importance of Objects

A web document gets transformed into a complex tree of objects.



A tree is defined as a collection of simple and complex objects.

# Types of Objects

BeautifulSoup transforms a complex HTML document into a complex tree of Python objects. There are four types of objects. They are:

Tag

A tag object is an XML or HTML tag in the web document. Tags have a lot of attributes and methods.

NavigableString

A NavigableString is a string or set of characters that corresponds to the text present within a tag.

BeautifulSoup

A BeautifulSoup represents the entire web document and supports navigating and searching the document tree.

Comment

A Comment represents the comment or information section of the document. It is a special type of NavigableString.



Demo: 01—Parsing web documents and extracting data using objects  
This demo shows you how to scrape a web document, parse it, and use objects to extract information.

DATA  
SCIENCE



# Knowledge Check

KNOWLEDGE  
CHECK

**Which of the following object types represents a string or set of characters within a tag?**

- a. Tag
- b. NavigableString
- c. BeautifulSoup
- d. Comment



KNOWLEDGE  
CHECK

**Which of the following object types represents a string or set of characters within a tag?**

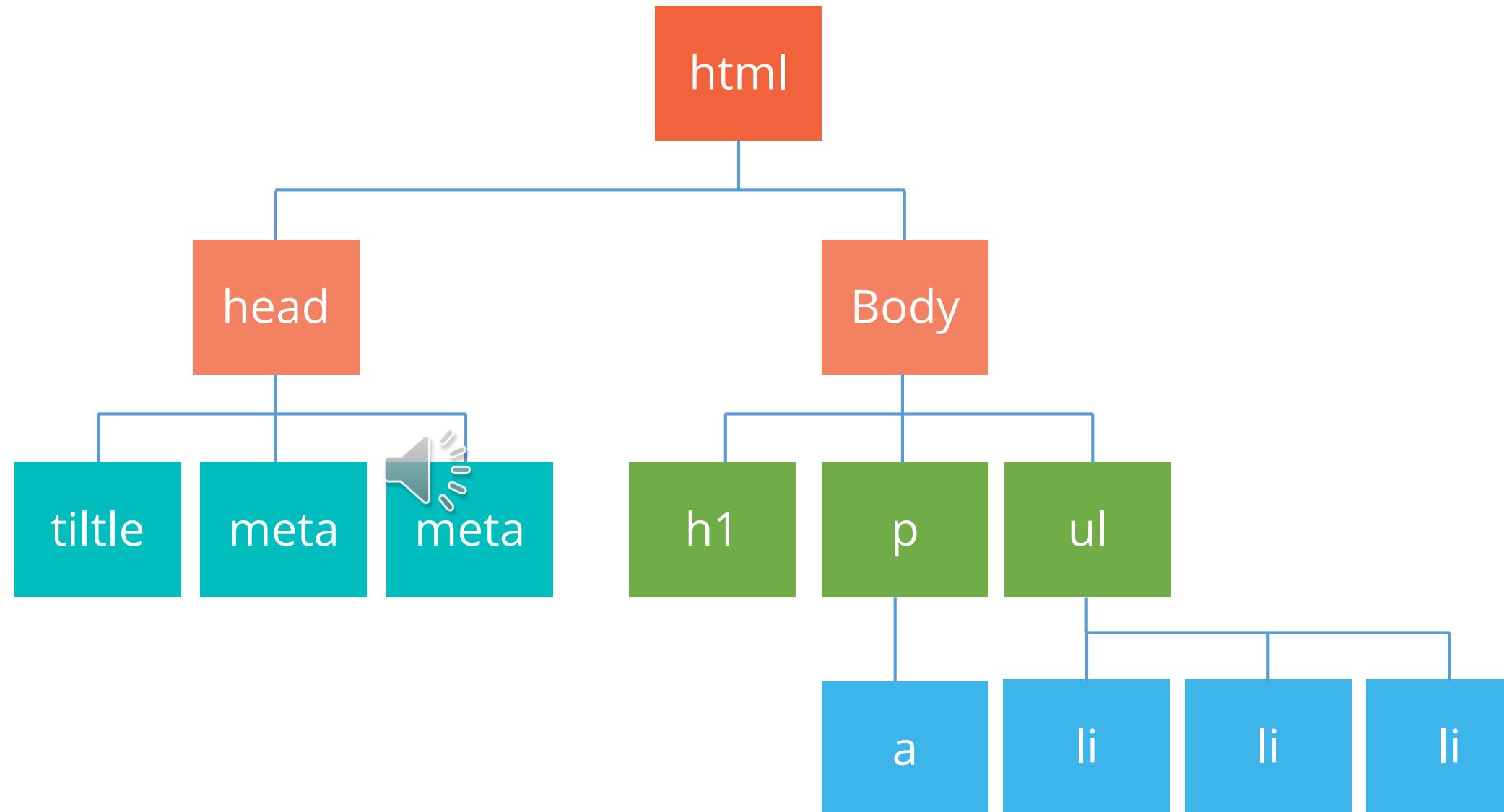
- a. Tag
- b. NavigableString
- c. BeautifulSoup
- d. Comment



The correct answer is . b.

Explanation: NavigableString is a string or set of characters that corresponds to the text present within a tag.

# Understanding the tree



# Understanding The Tree

```
<!DOCTYPE html>
<html>
  <body>
    <div class="oraganizationlist">
      <ul id="HR">
        <li class="HRmanager">
          <div class="name">Jack</div>
          <div class="ID">101</div>
        </li>
        <li class="HRmanager">
          <div class="name">Daren</div>
          <div class="ID">65</div>
        </li>
      </ul>
      <ul id="IT">
        <li class="ITmanager">
          <div class="name">Morris</div>
          <div class="ID">39</div>
        </li>
        <li class="ITmanager">
          <div class="name">Jane</div>
          <div class="ID">11</div>
        </li>
      </ul>
      <ul id="Finance">
        <li class="accountmanager">
          <div class="name">Tom</div>
          <div class="ID">22</div>
        </li>
        <li class="accountmanager">
          <div class="name">Kelly</div>
          <div class="ID">95</div>
        </li>
      </ul>
    </body>
  </html>
```

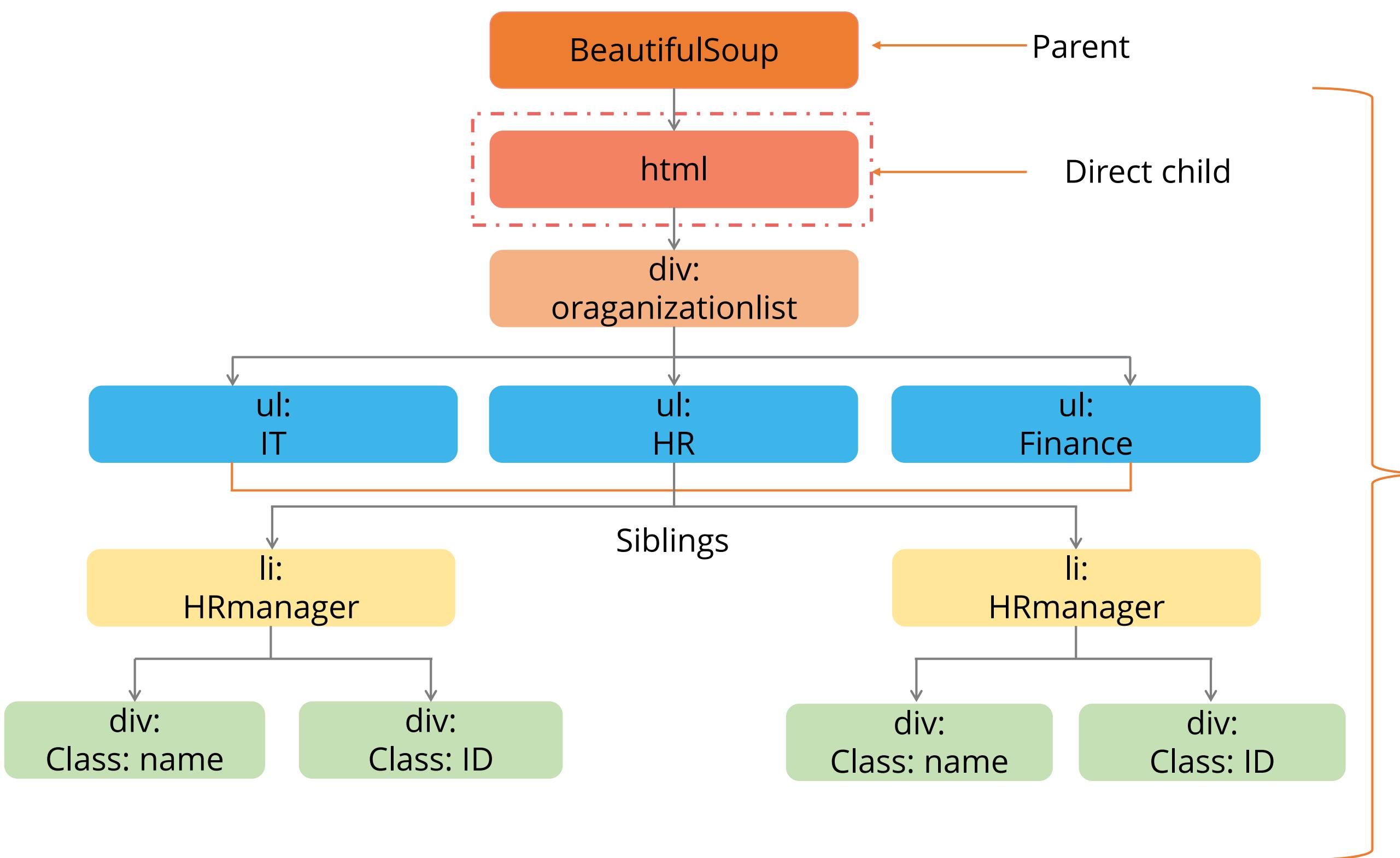
html tag

Body tag

Division or a Section

Cascaded style sheets

# Understanding The Tree (contd.)



# Searching The Tree – Filters

With the help of the search filters technique, you can extract specific information from the parsed document.

The filters can be treated as search criteria for extracting the information based on the elements present in the document.



## Searching The Tree – Filters (contd.)

There are various kinds of filters used for searching an information from a tree:

String

A string is the simplest filter. BeautifulSoup will perform a match against the search string.

Regular  
Expressions

A regular expression filters the match against the search criteria.

List

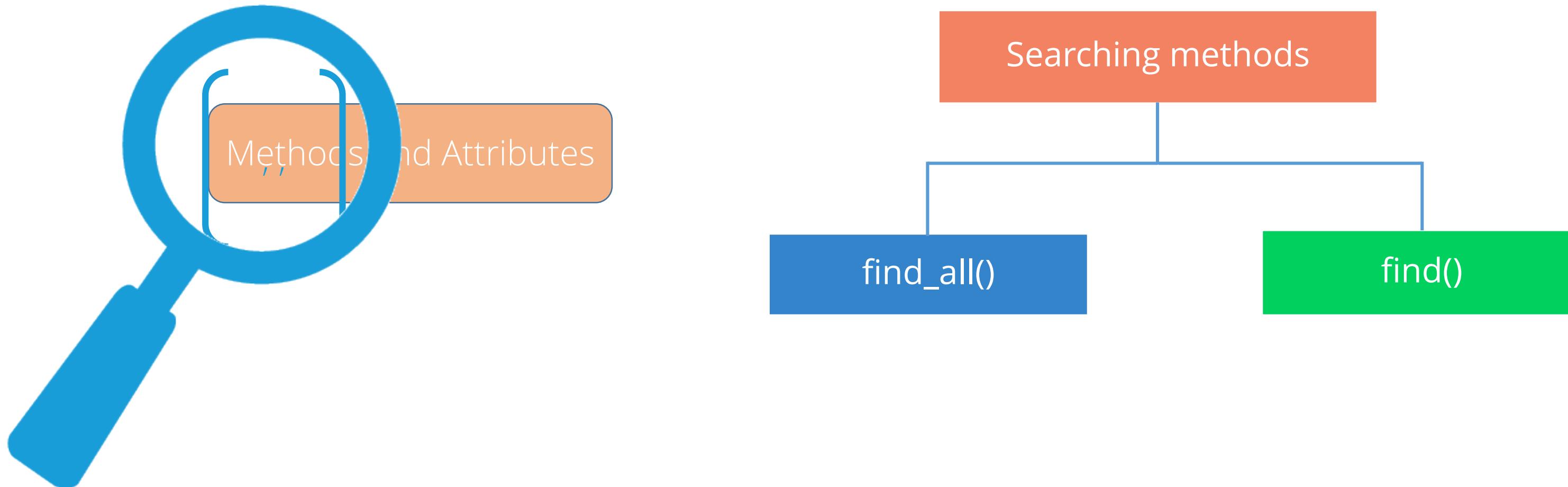
A list filters the string that matches against the search item in the list.

Function

A function filters the elements that matches against its only argument.

# Searching the Tree—`find_all()`

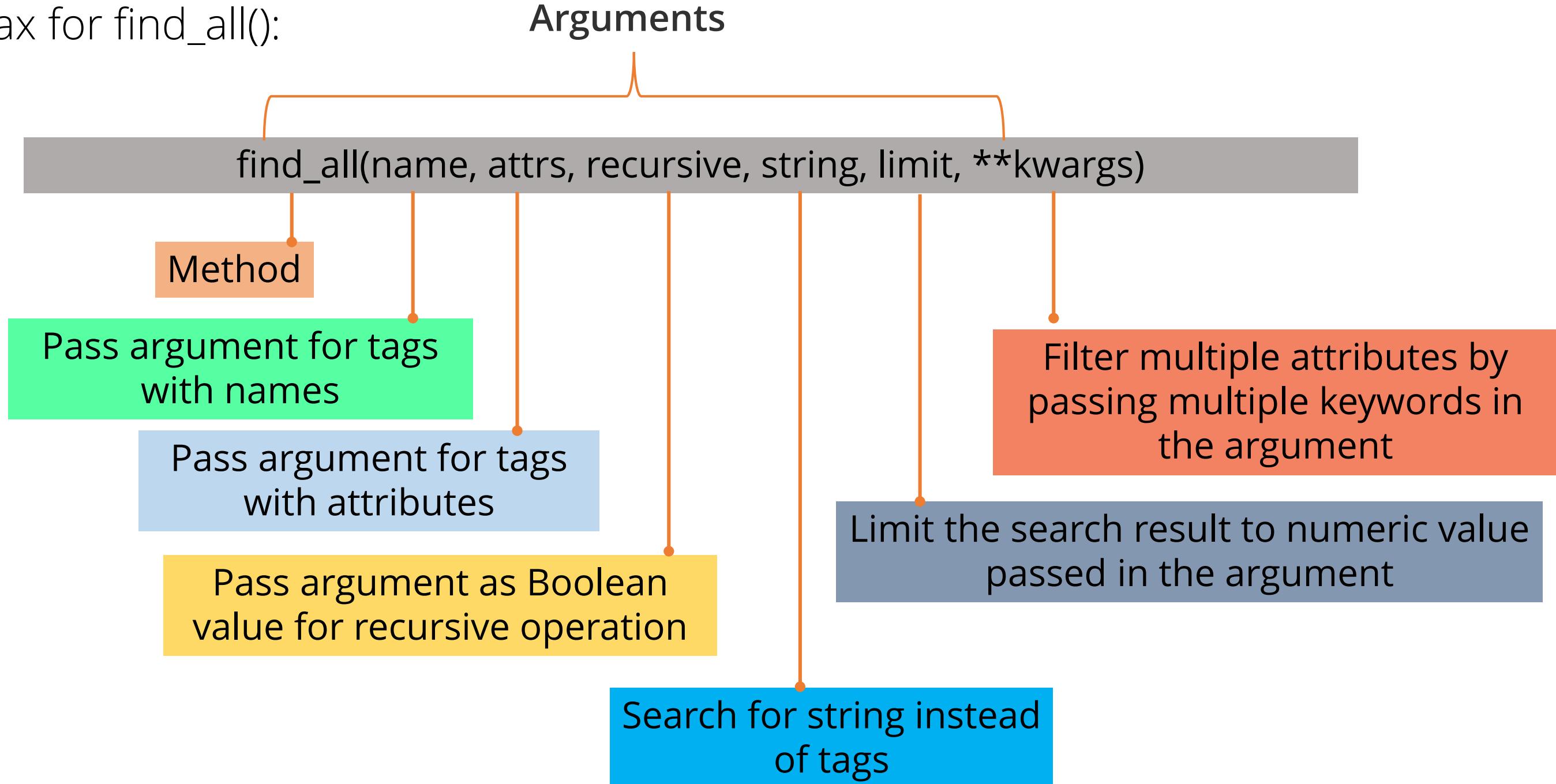
BeautifulSoup defines a lot of methods for searching the parsed tree.



# Searching the tree with find\_all()

The find\_all() searches and retrieves all tags' descendants that matches your filters.

The syntax for find\_all():



# Searching the tree with find ()

The find\_all() finds the entire document looking for results.

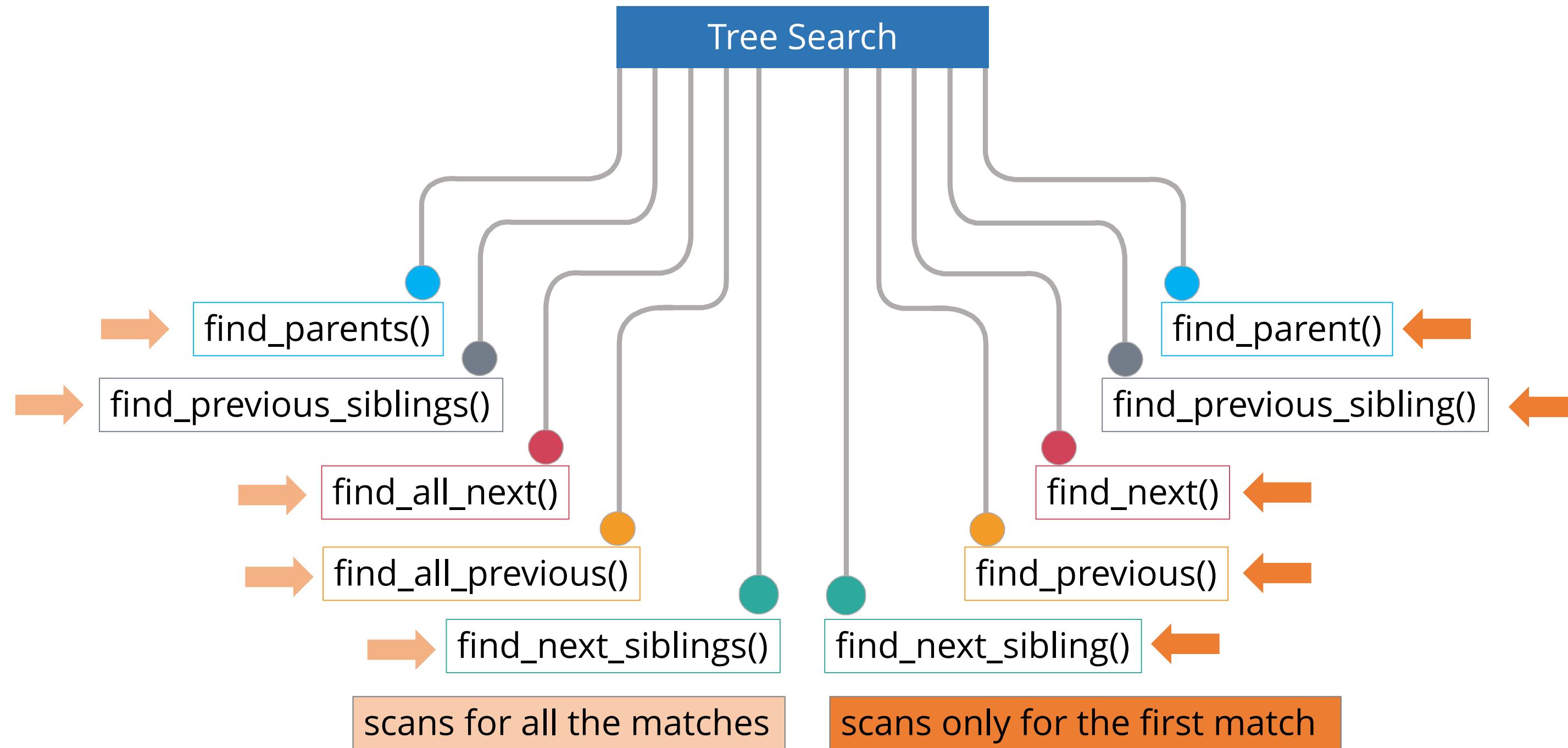
To find one result, use find().

The find() method has a syntax similar to that of the find\_all() method; however, there are some key differences.

Method name	Search Scope	Match Found	Match Not Found
Find_all()	Scans entire document	Returns list with values	Returns empty list
Find()	Searches only for passed argument	Returns only the first match value	Returns None

# Searching the tree with other methods

Searching the parse tree can also be performed by various other methods such as the following:





Demo: 02—Demo: 02—Searching in a Tree with Filters  
This demo shows the ways to search in a tree using filters.



# Knowledge Check

KNOWLEDGE  
CHECK

**The method `get_text()` is used to \_\_\_\_\_.**

- a. parse the entire document
- b. parse only part of the document
- c. search the tree
- d. navigate the tree



KNOWLEDGE  
CHECK

**The method `get_text()` is used to \_\_\_\_\_.**

- a. parse the entire document
- b. parse only part of the document
- c. search the tree
- d. navigate the tree



The correct answer is . b.

Explanation The method `get_text()` is used to parse only part of the document.

# Navigating options

---

With the help of BeautifulSoup, it is easy to navigate the parse tree based on the need.

There are four options to navigate the tree:

*Click each tab to know more.*

Navigating Down

Navigating Up

Navigating  
Sideways

Navigating Back  
and Forth

# Navigating options

With the help of BeautifulSoup, it is easy to navigate the parse tree based on the need.

There are four options to navigate the tree. They are:

*Click each tab to know more.*

Navigating Down

This technique shows you how to extract information from children tags. Following are the attributes used to navigate down:

Navigating Up

- .contents and .children
- .descendants
- .string
- .strings and stripped\_strings

Navigating Sideways

Navigating Back  
and Forth

# Navigating options

With the help of BeautifulSoup, it is easy to navigate the parse tree based on the need.

There are four options to navigate the tree:

*Click each tab to know more.*

Navigating Down

Navigating Up

Navigating  
Sideways

Navigating Back  
and Forth

Navigating Up:

Every tag has a parent and two attributes, .parents and .parent,to help navigate up the family tree.

# Navigating options

With the help of BeautifulSoup, it is easy to navigate the parse tree based on the need.

There are four options to navigate the tree:

*Click each tab to know more.*

Navigating Down

Navigating Up

Navigating  
Sideways

Navigating Back  
and Forth

Navigating Sideways:

This technique shows you how to extract information from the same level in the tree.

The attributes used to navigate sideways are `.next_sibling` and `.previous_sibling`.

# Navigating options

With the help of BeautifulSoup, it is easy to navigate the parse tree based on the need.

There are four options to navigate the tree:

*Click each tab to know more.*

Navigating Down

Navigating Up

Navigating  
Sideways

Navigating Back  
and Forth

Navigating Back and Forth:

This technique shows you how to parse the tree back and forth.  
Following are the attributes used to navigate back and forth are:  
.next\_element and .previous\_element  
.next\_elements and .previous\_elements



## Demo: 03—Navigating a Tree

This demo shows how to navigate the web tree using various techniques.

DATA SCIENCE



# Knowledge Check

KNOWLEDGE  
CHECK

**Which of the following attributes is used to navigate up?**

- a. .next\_element
- b. .parent
- c. .previous\_elements
- d. .next\_sibling



KNOWLEDGE  
CHECK**Which of the following attributes is used to navigate up?**

- a. .next\_element
- b. .parent
- c. .previous\_elements
- d. .next\_sibling



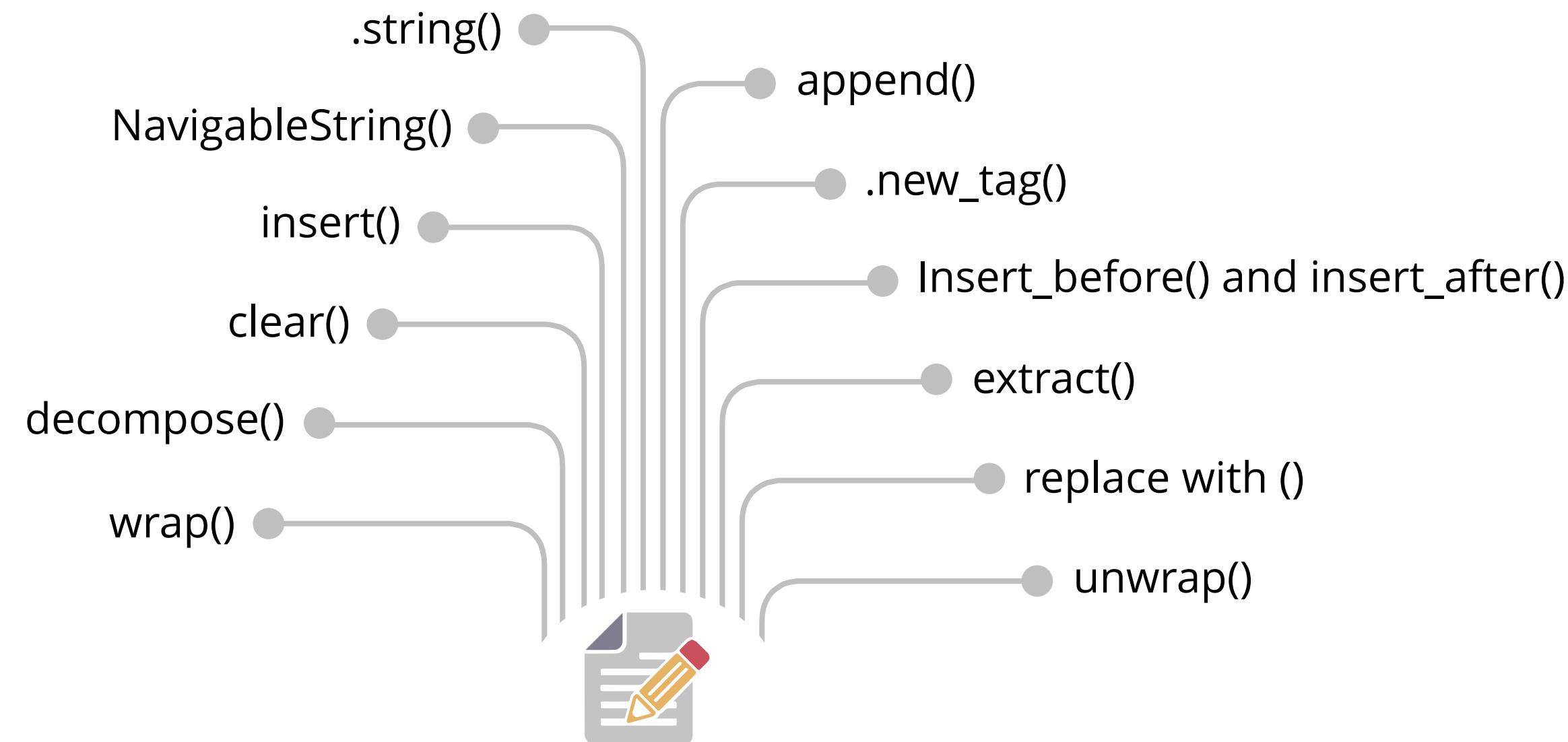
The correct answer is . b.

Explanation: The .parent attribute is used to navigate up.

# Modifying The Tree

With BeautifulSoup, you can also modify the tree and write your changes as a new HTML or XML document.

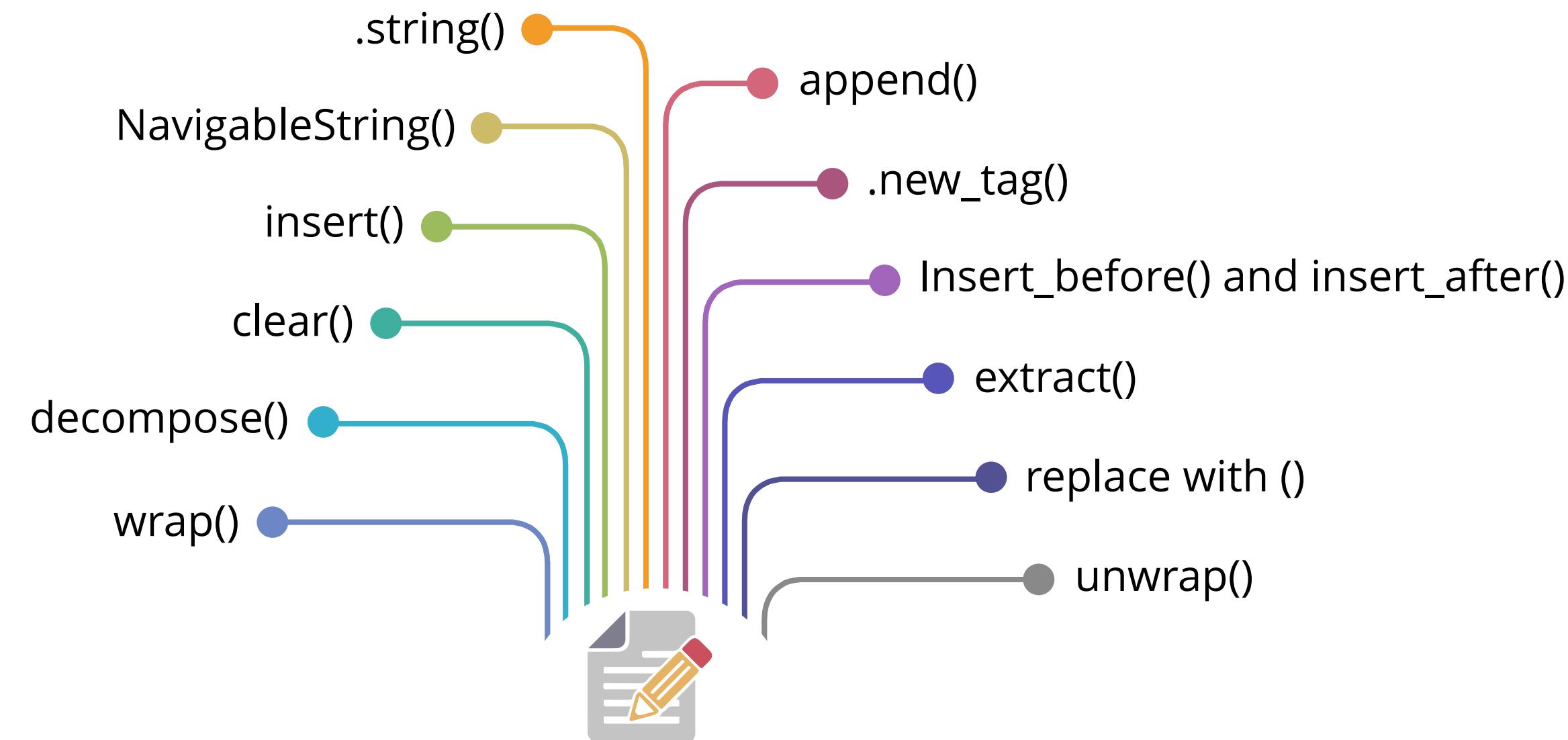
There are several methods to modify the tree:



# Modifying The Tree

With BeautifulSoup, you can also modify the tree and write your changes as a new HTML or XML document.

There are several methods to modify the tree:

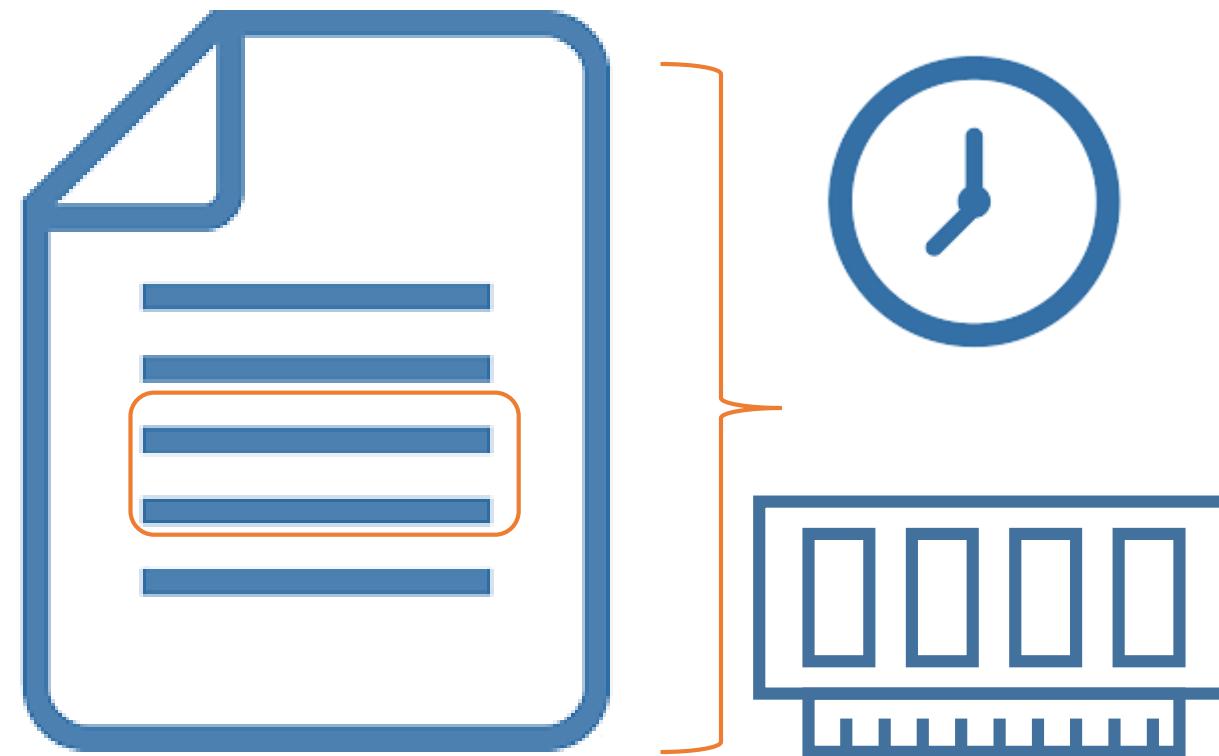




## Demo: 04—Modifying the Tree

This demo shows you ways to modify a web tree to get the desired result with the help of an example.

# Parsing Only Part of the Document



But how can you overcome this problem?

Use `SoupStrainer` class

Allows you to choose the part of the document to be parsed



This feature of parsing a part of the document will not work with the `html5lib` parser.

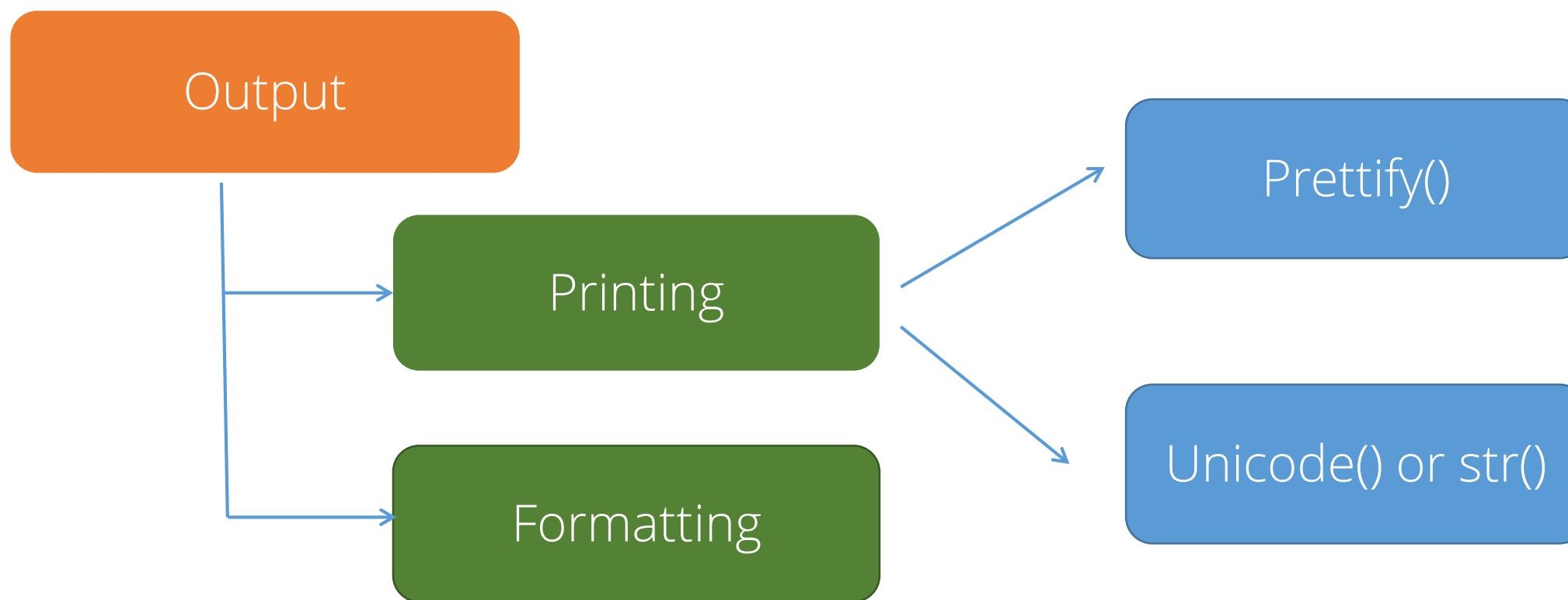


## Demo: 05—Parsing part of the document

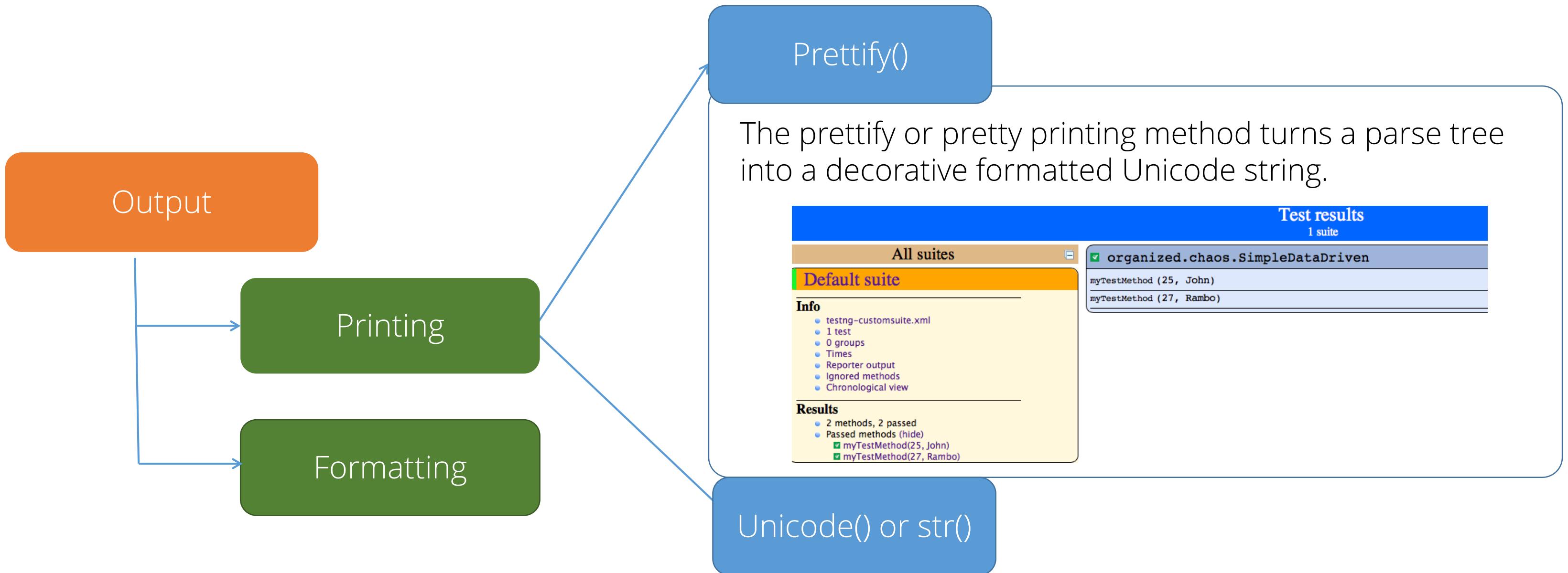
This demo shows you how to parse only a part of document with the help of an example.

DATA SCIENCE

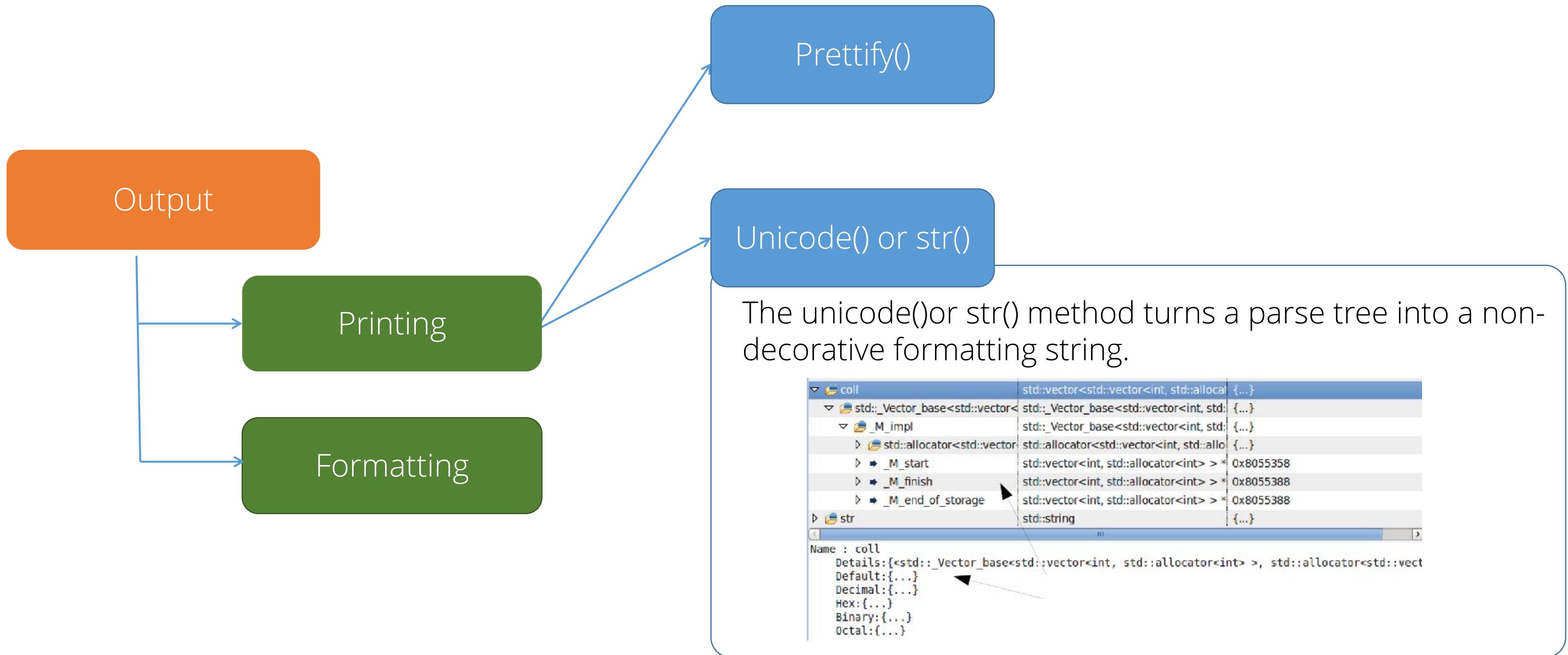
# Output : Printing and Formatting



# Output : Printing and Formatting

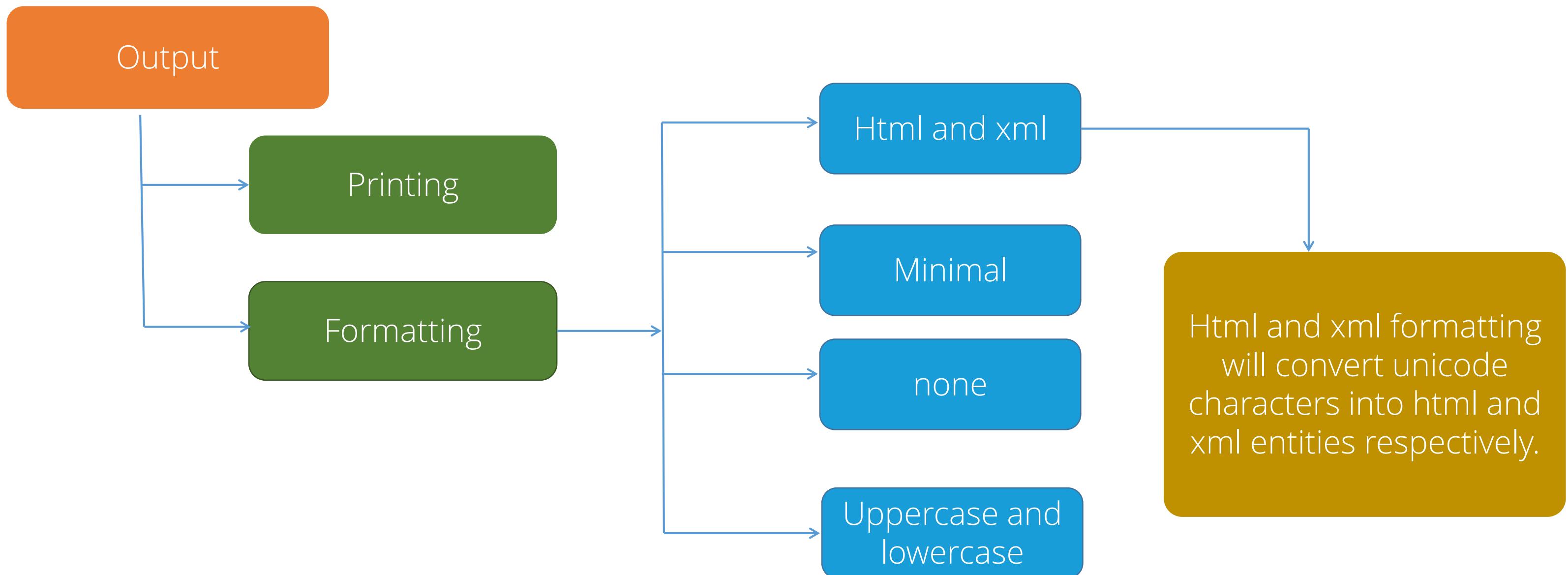


# Output : Printing and Formatting (contd.)



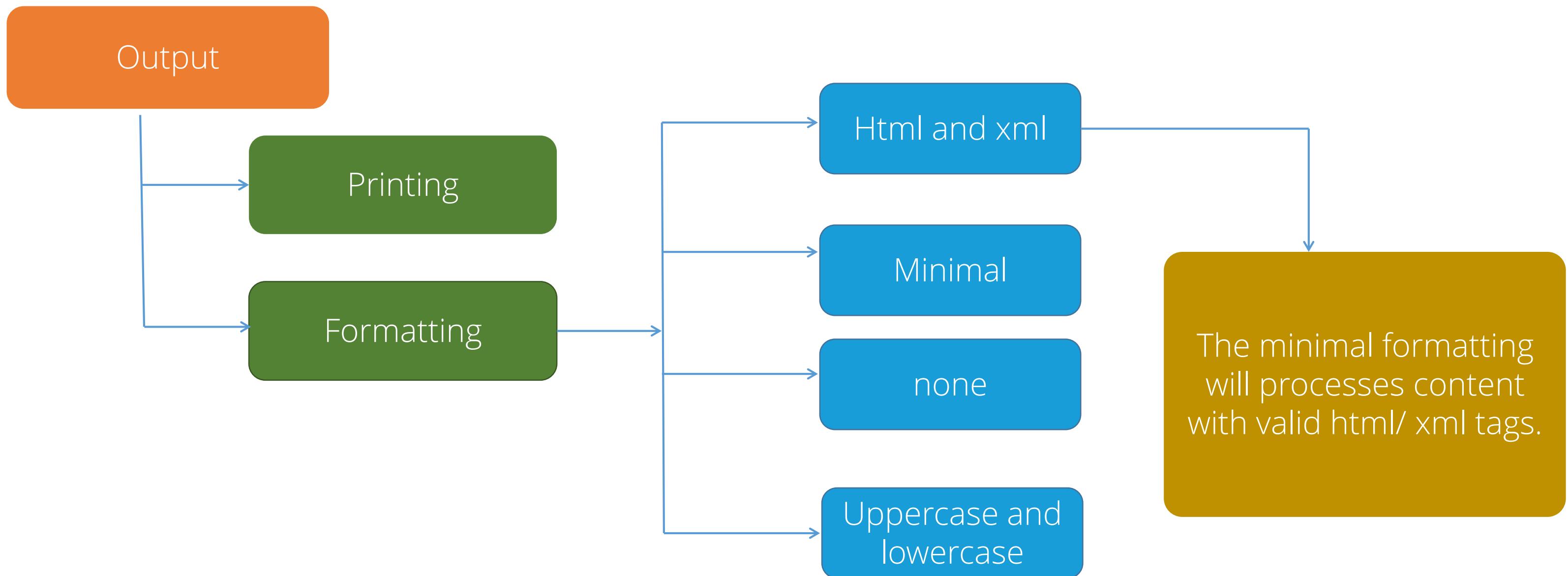
# Output : Printing and Formatting (contd.)

The formatters are used to generate different types of output with the desired formatting.



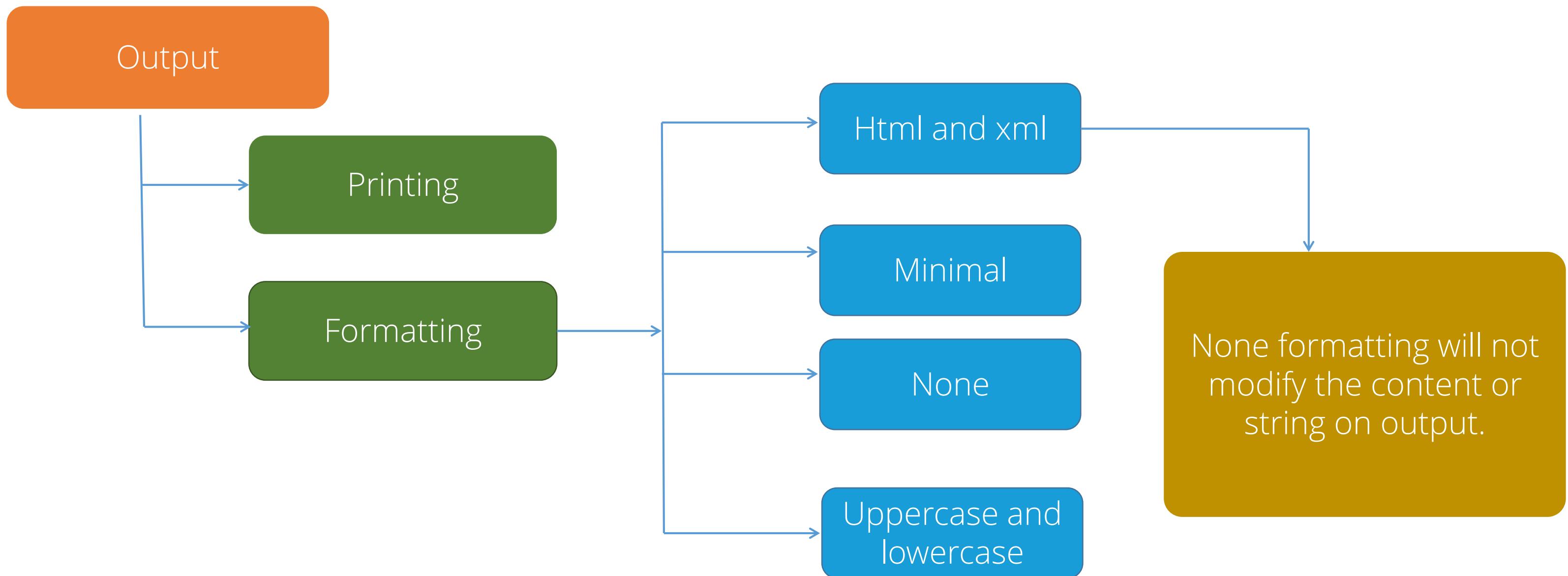
# Output : Printing and Formatting (contd.)

The formatters are used to generate different types of output with the desired formatting.



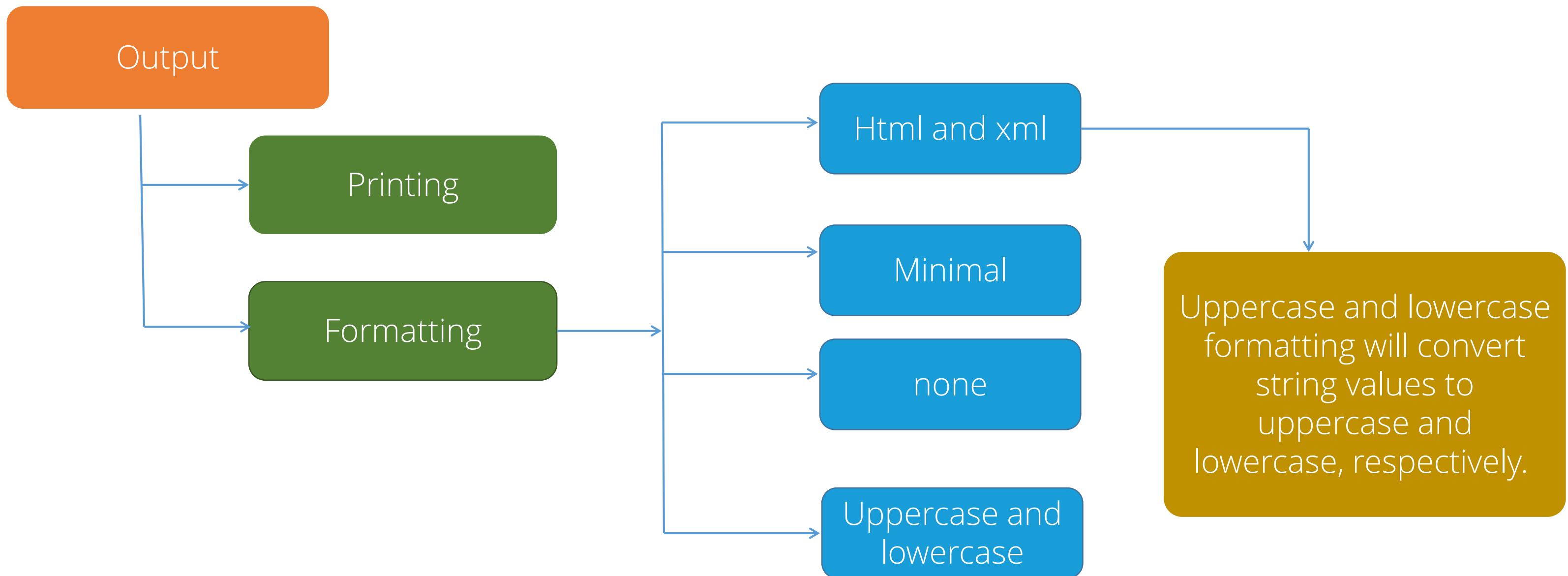
# Output : Printing and Formatting (contd.)

The formatters are used to generate different types of output with the desired formatting.



# Output : Printing and Formatting (contd.)

The formatters are used to generate different types of output with the desired formatting.





## Demo: 06—Formatting and Printing

This demo shows the ways to format, print, and encode the web document.

## Document Encoding

- HTML or XML documents are written in specific encodings such as, ASCII or UTF-8.
- When we load the document into BeautifulSoup, it gets converted into Unicode.
- The original encoding can be extracted from attribute .original\_encoding of the Beautiful Soup object.

## Output Encoding

- When you write a document from BeautifulSoup, you get a UTF-8 document irrespective of the original encoding.
- If some other encoding is required, we can pass it to prettify.



# Knowledge Check

Problem

Instructions

Scrape the Simplilearn website page and perform the following tasks:

- View and print the Simplilearn web page content in a proper format
- View the head and title
- Print all the href links present in the Simplilearn web page

Simplilearn website URL: <http://www.simplilearn.com/>

Problem

Instructions

Instructions to perform the assignment:

- Use Simplilearn's resource page URL in the Jupyter notebook to view and evaluate it.

Common instructions:

- If you are new to Python, download the "Anaconda Installation Instructions" document from the "Resources" tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the "Assignment 01" notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



# Knowledge Check

Problem

Instructions

Scrape the Simplilearn website resource page and perform the following tasks:

- View and print the Simplilearn web page content in a proper format
- View the head and title
- Print all the href links present in the Simplilearn web page
- Search and print the resource headers of the Simplilearn web page
- Search resource topics
- View the article names and navigate through them

Simplilearn website URL: <http://www.simplilearn.com/resources>

Problem

Instructions

Instructions to perform the assignment:

- Download the web scraping dataset from the “Resource” tab. Upload the dataset to your Jupyter notebook to view and evaluate it.

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



**QUIZ****1**

**Which of the following is the only xml parser?**

- a. html.parser
- b. lxml
- c. lxml.xml
- d. html5lib



**QUIZ****1**

**Which of the following is the only xml parser?**

- a. html.parser
- b. lxml
- c. lxml.xml
- d. html5lib



The correct answer is .

**Explanation: lxml.xml is the only xml parser available for BeautifulSoup object.**

**QUIZ**  
**2**

**In which of the following formats is the BeautifulSoup output encoded?**

- a. ASCII
- b. Unicode
- c. latin-1
- d. UTF-8



**QUIZ**  
**2**

**In which of the following formats is the BeautifulSoup output encoded?**

- a. ASCII
- b. Unicode
- c. latin-1
- d. UTF-8



The correct answer is **d**.

**Explanation: The output of the BeautifulSoup is always UTF-8 encoded.**

**QUIZ****3**

**Which of the following libraries is used to extract a web page?**

- a. Beautiful Soup
- b. Pandas
- c. Requests
- d. NumPy



**QUIZ**  
**3**

**Which of the following libraries is used to extract a web page?**

- a. Beautiful Soup
- b. Pandas
- c. Requests
- d. NumPy



The correct answer is **c**.

**Explanation: Requests is the right API to extract the web page.**

**QUIZ****4**

**Which of the following is NOT an object in BeautifulSoup?**

- a. Tag
- b. NextSibling
- c. NavigableString
- d. Comment



## QUIZ

4

Which of the following is NOT an object in BeautifulSoup?

- a. Tag
- b. Next sibling
- c. NavigableString
- d. Comment

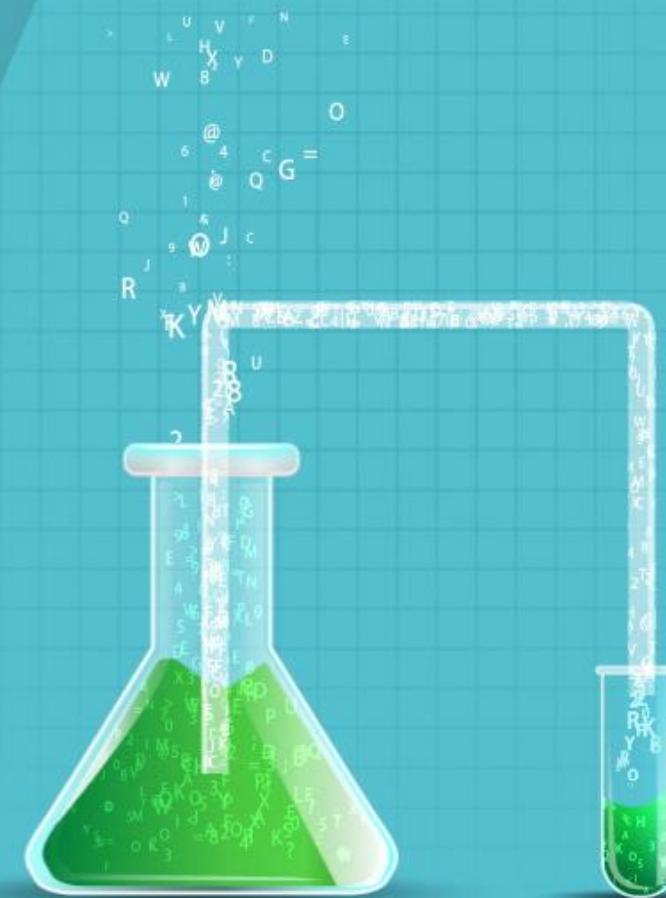


The correct answer is **b**.

**Explanation:** NextSibling is a navigation method.

# Key Takeaways

- Web scraping is a computer software technique of extracting information from websites in an automated fashion.
- A Parser is a basic tool to interpret or render the information from a web document.
- Objects are used to extract the required information from a tree structure by searching or navigating through the parsed document.
- A tree can be defined as a collection of simple and complex objects.
- BeautifulSoup transforms a complex HTML document into a complex tree of Python objects.



**This concludes “Web Scraping with BeautifulSoup”**

The next lesson is “Python integration with Hadoop, MapReduce, and Spark”

DATA  
SCIENCE

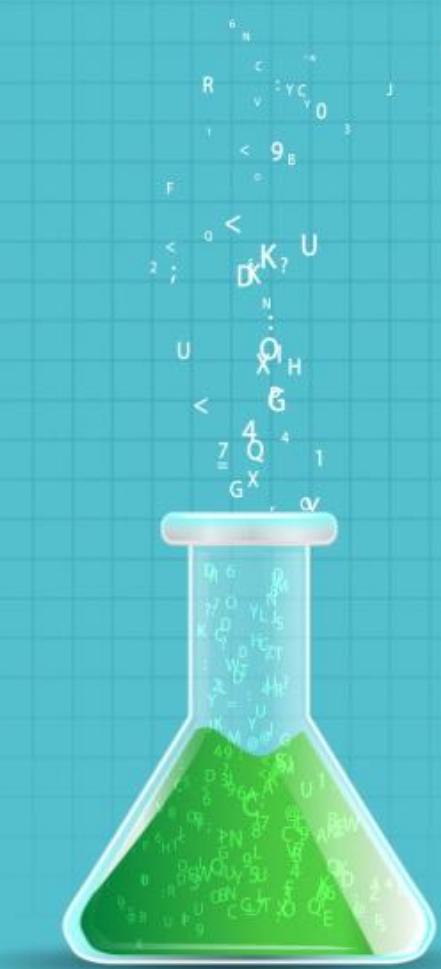
# Data Science with Python

## Lesson 12—Python Integration with Hadoop MapReduce and Spark

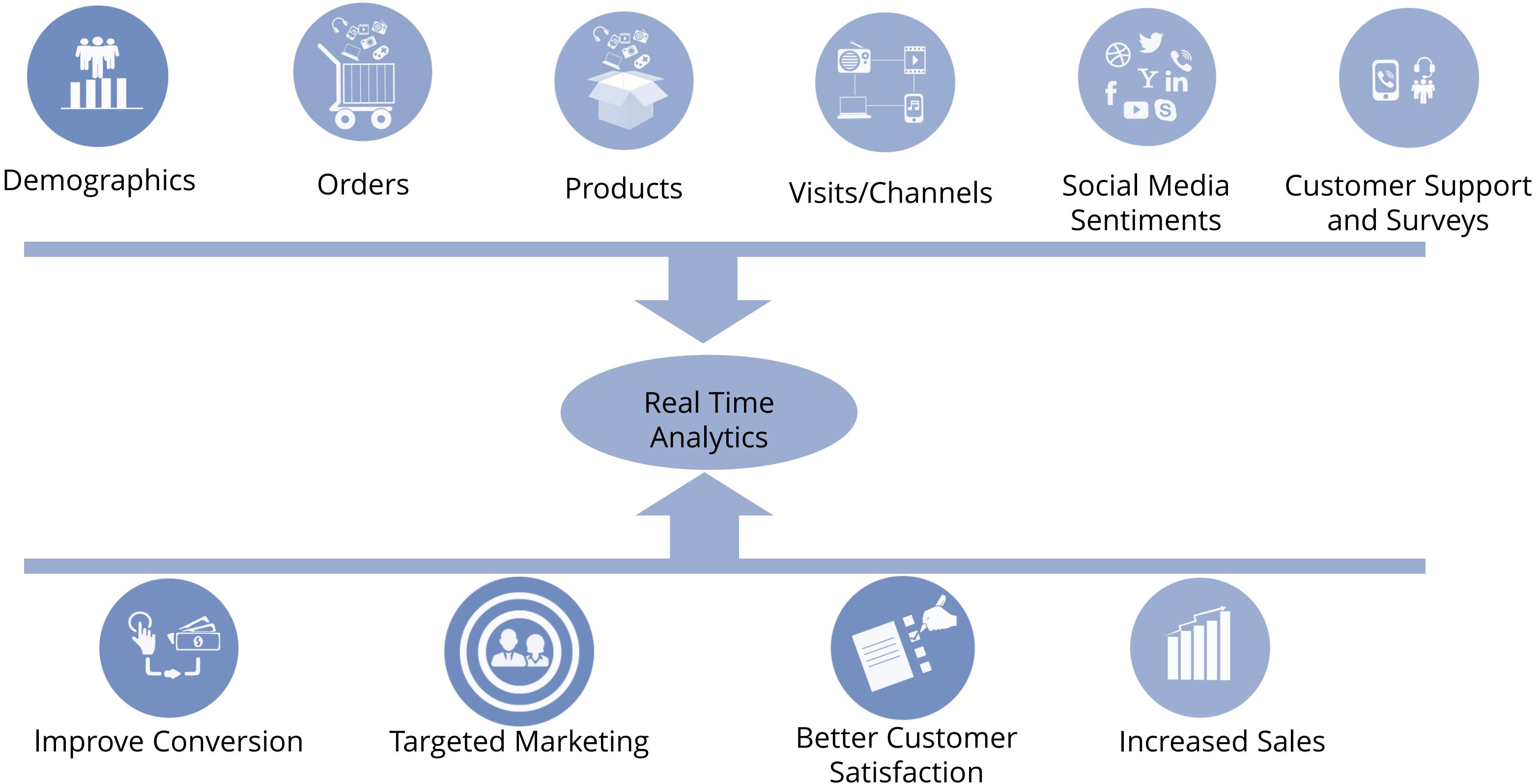


# What You'll Learn

- Why Python should be integrated with Hadoop
- Brief overview of the ecosystem and architecture of Hadoop
- How MapReduce functions
- How Apache Spark functions and what its benefits are
- Write Python programs for Hadoop operations

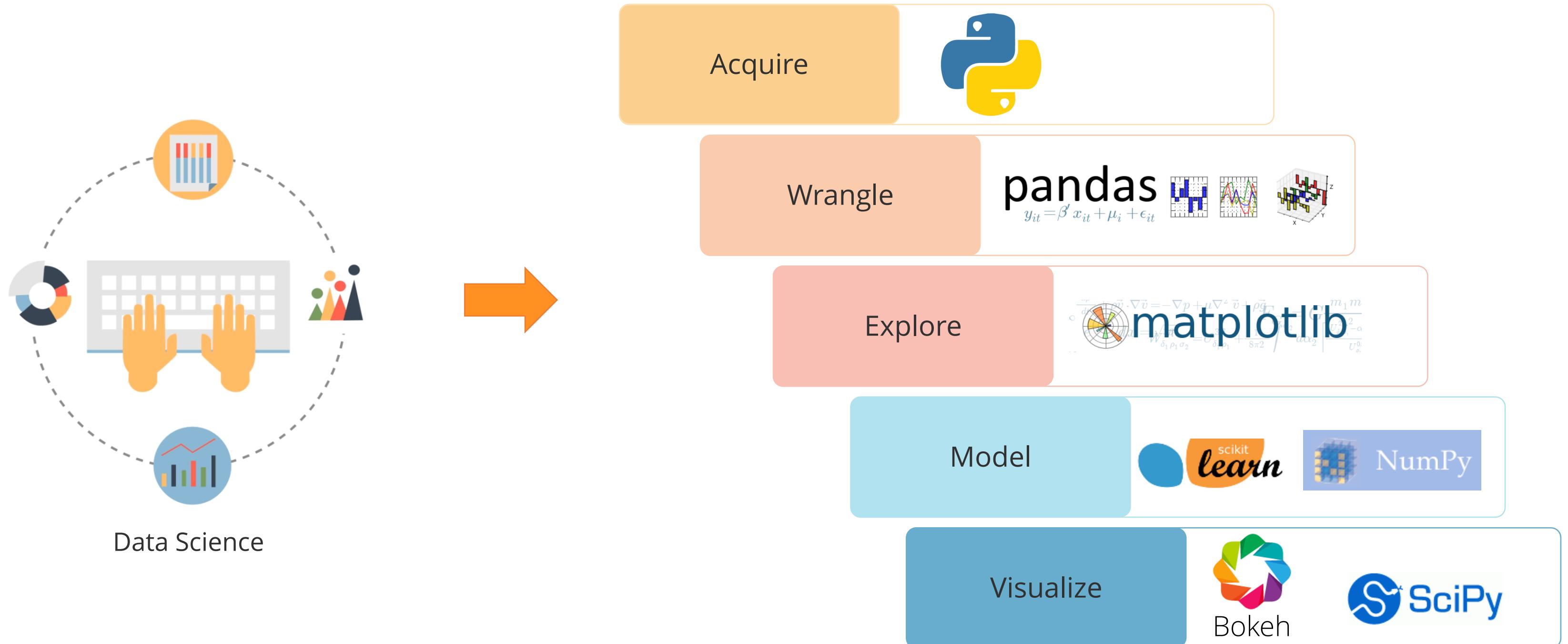


# Quick Recap: Need for Real Time Analytics



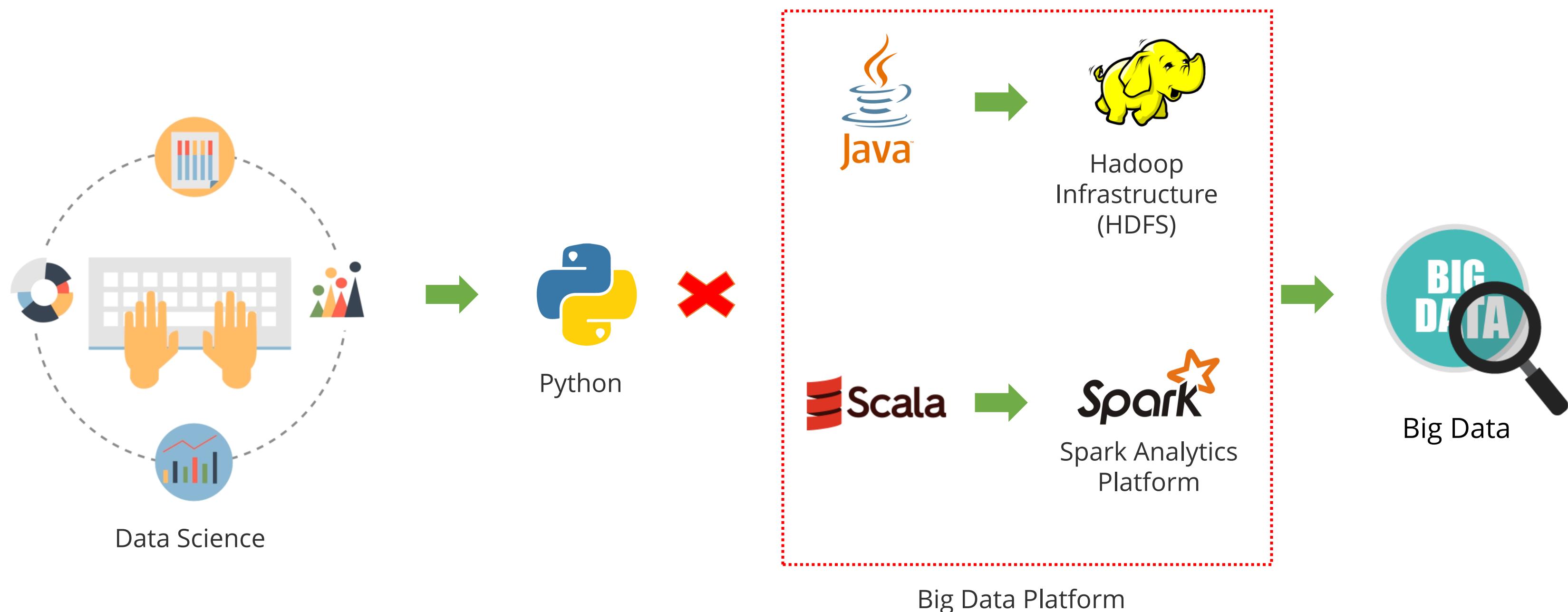
# Quick Recap: Why Python

Data Scientists all over the world prefer to use Python for analytics because of its ease and support to carry out all the aspects of Data Science.



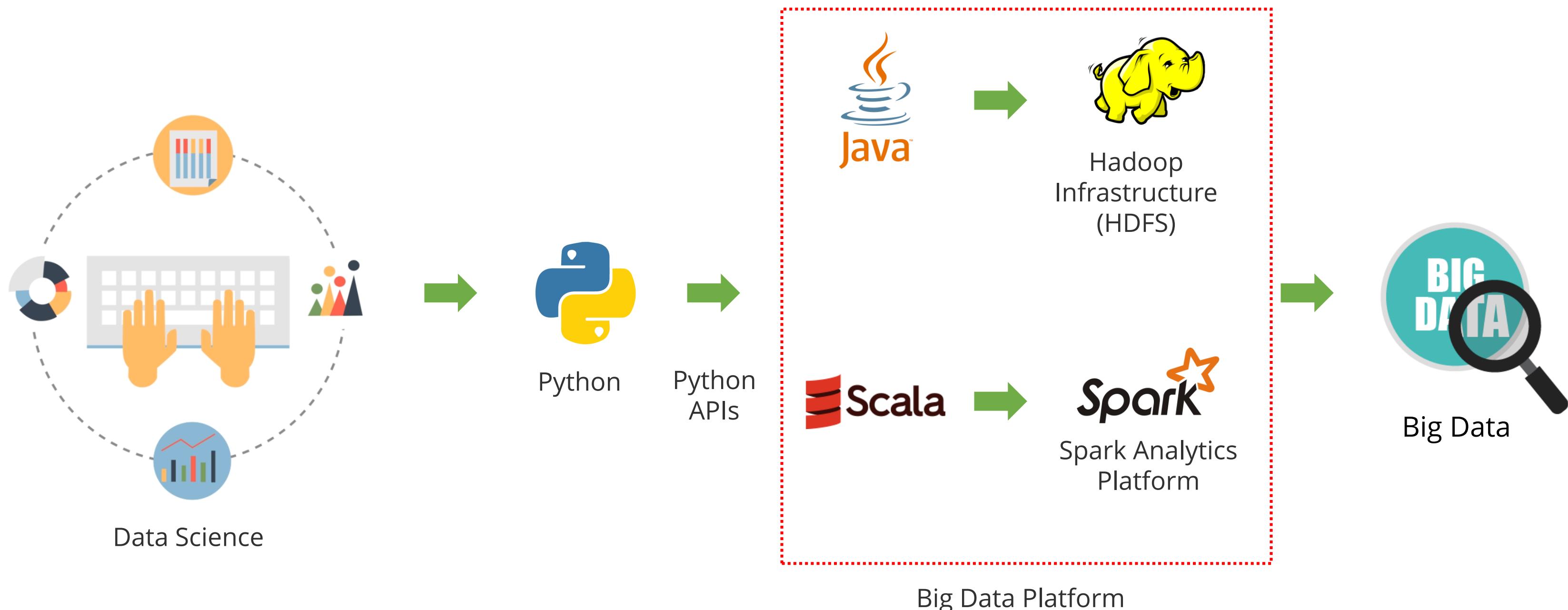
# Disparity in Programming Languages

However, Big Data can only be accessed through Hadoop which is completely developed and implemented in Java. Also, analytics platforms are coded in different programming languages.



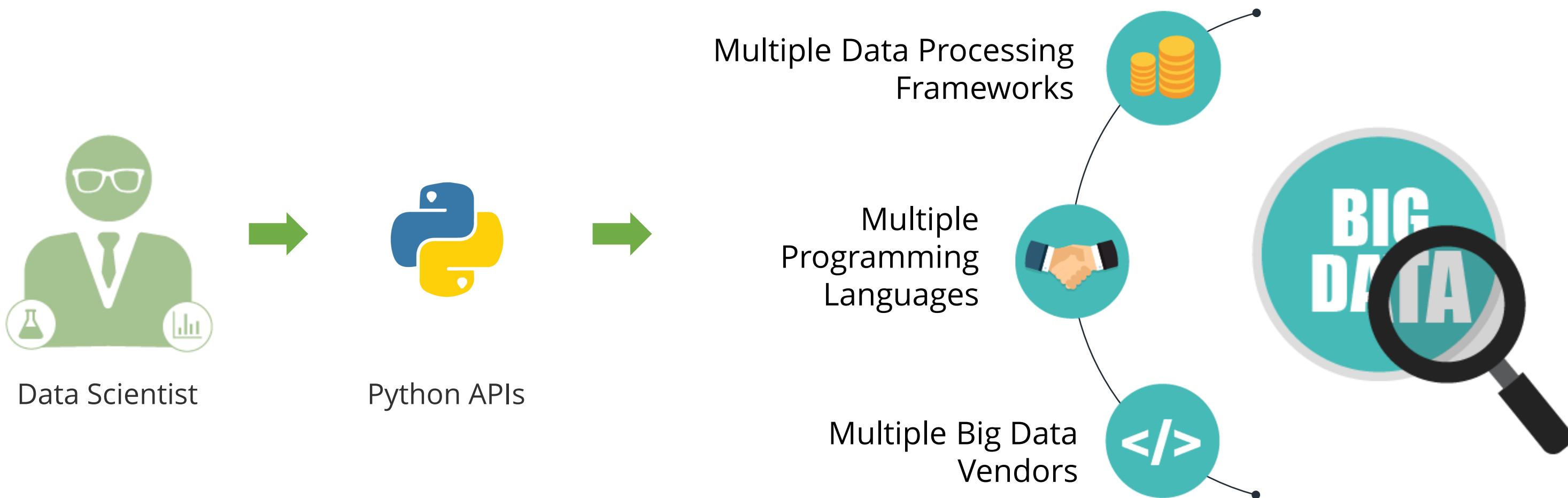
# Integrating Python with Hadoop

But as Python is a Data Scientist's first language of choice, both Hadoop and Spark provide Python APIs that allow easy access to the Big Data platform.

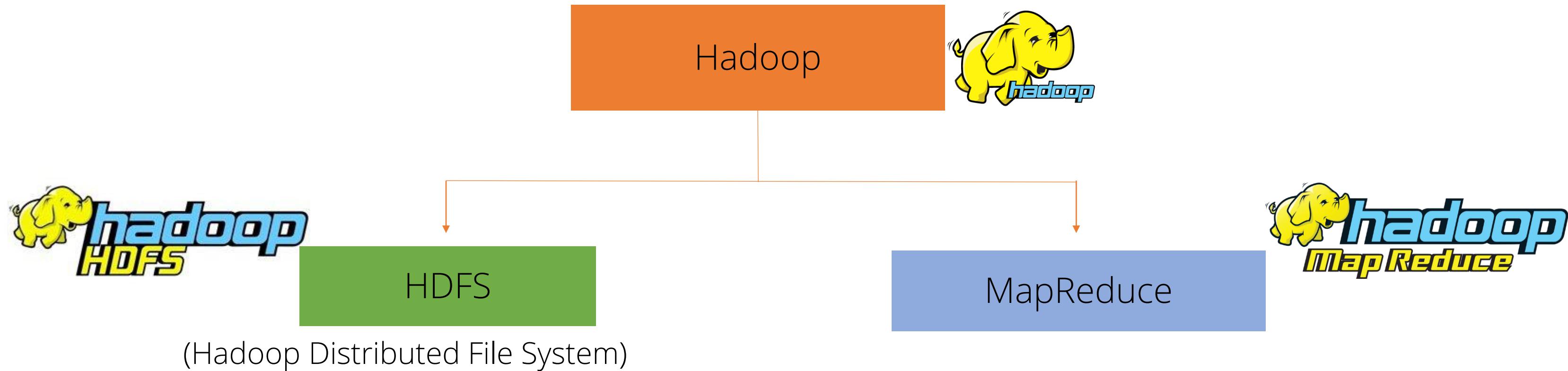


# Need for Big Data Solutions for Python

There are several reasons for creating Big Data solutions for Python.



# Hadoop: Core Components

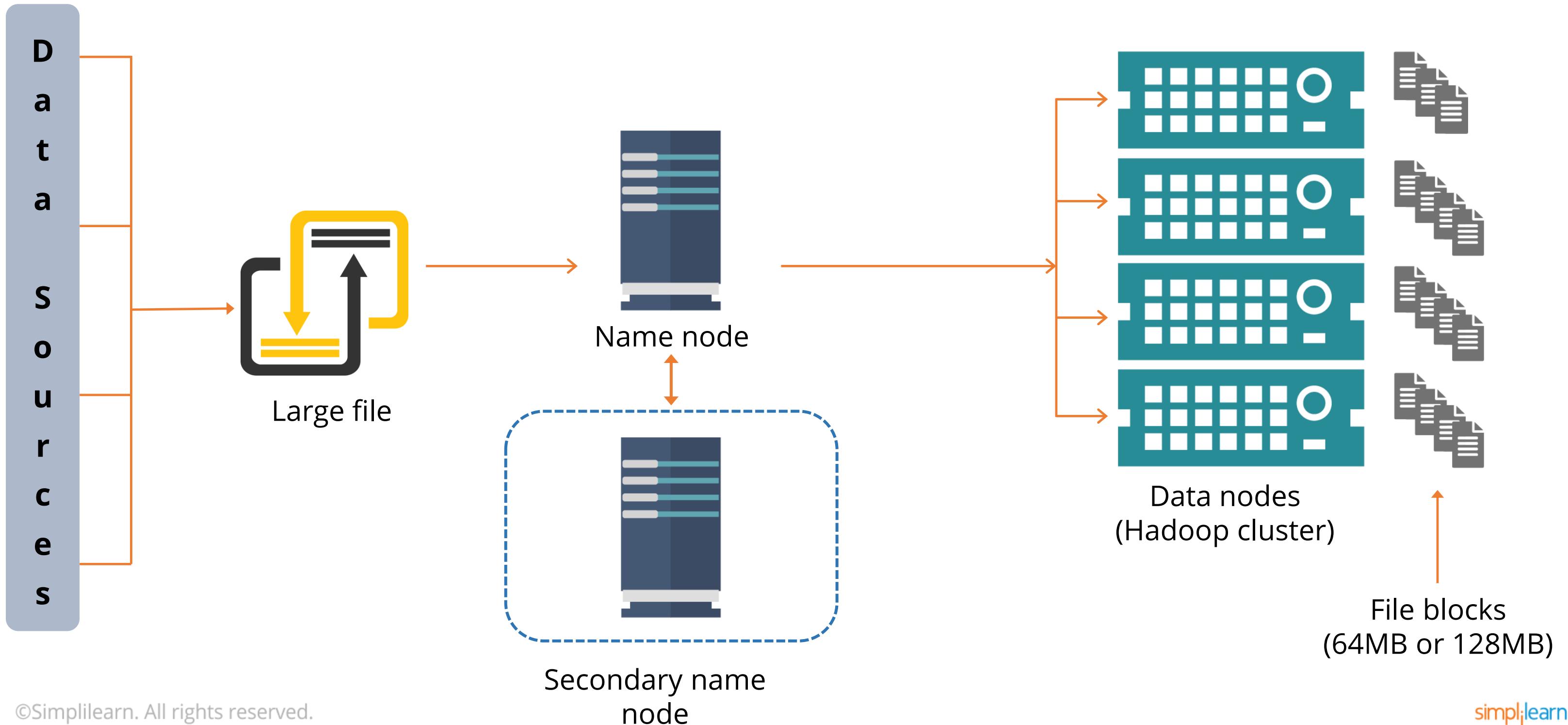


- It is responsible for storing data on a cluster
- Data is split into blocks and distributed across multiple nodes in a cluster
- Each block is replicated multiple times
  - Default is 3 times
  - Replicas are stored on different nodes

- MapReduce is a data processing framework to process data on the cluster
- Two consecutive phases: Map and Reduce
- Each map task operates on discrete portions of data
- After map, reduce works on the intermediate data distributed on nodes

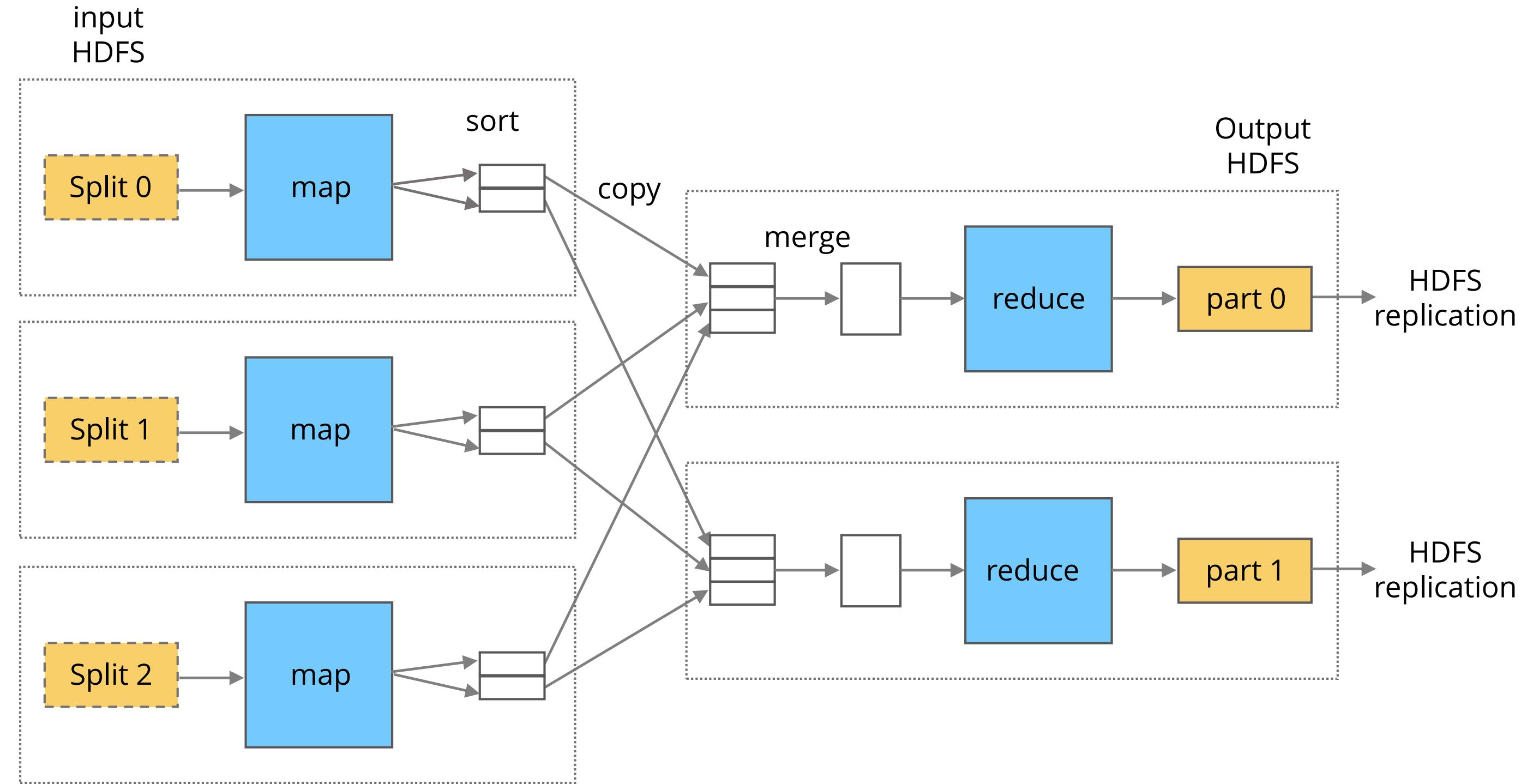
# Hadoop: The System Architecture

This example illustrates the Hadoop system architecture and the ways to store data in a cluster.



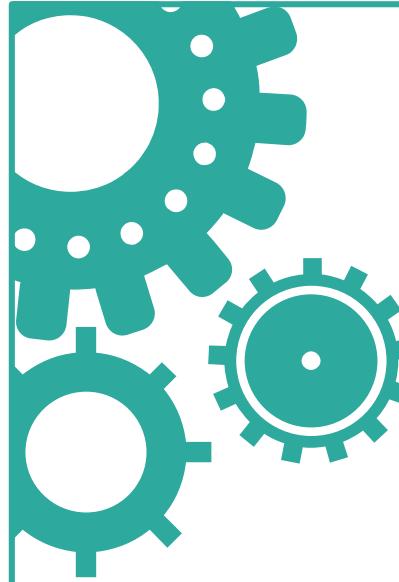
# MapReduce

The second core component of Hadoop is MapReduce, the primary framework of the HDFS architecture.



# **MapReduce: The Mapper and Reducer**

Let us discuss the MapReduce functions—mapper and reducer—in detail.



## **Mapper**

- Mappers run locally on the data nodes to avoid the network traffic.
- Multiple mappers run in parallel processing a portion of the input data.
- The mapper reads data in the form of key-value pairs.
- If the mapper writes generates an output, it is written in the form of key-value pairs.

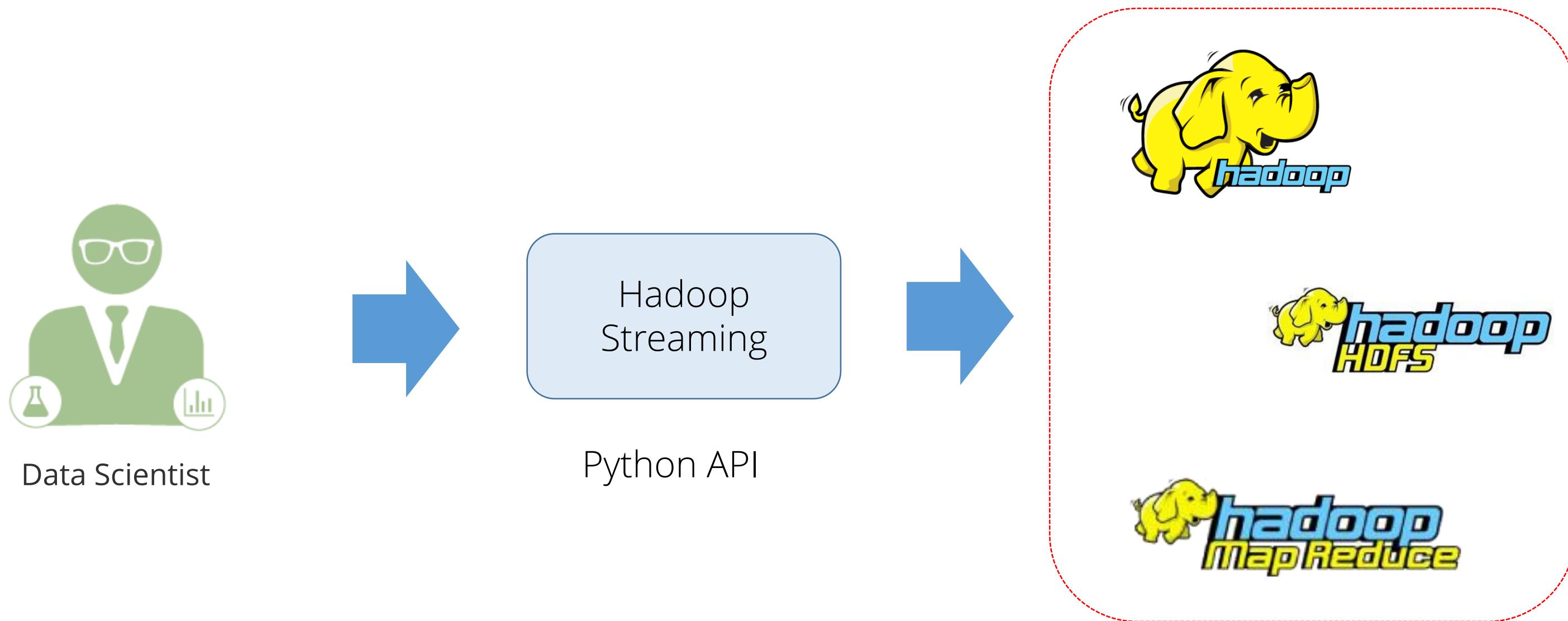
## **Reducer**

- All intermediate values for a given intermediate key are combined together into a list and given to a reducer.
- This step is known as 'shuffle and sort'.
- The reducer outputs either zero or more final key-value pairs. These are written to HDFS.



# Hadoop Streaming: Python API for Hadoop

Hadoop Streaming acts like a bridge between your Python code and the Java-based HDFS, and lets you seamlessly access Hadoop clusters and execute MapReduce tasks.



# Mapper in Python

Python supports map and reduce operations:

Suppose you have list of numbers you want to square =  
[1, 2, 3, 4, 5, 6 ]

**Square** function is written as follows:

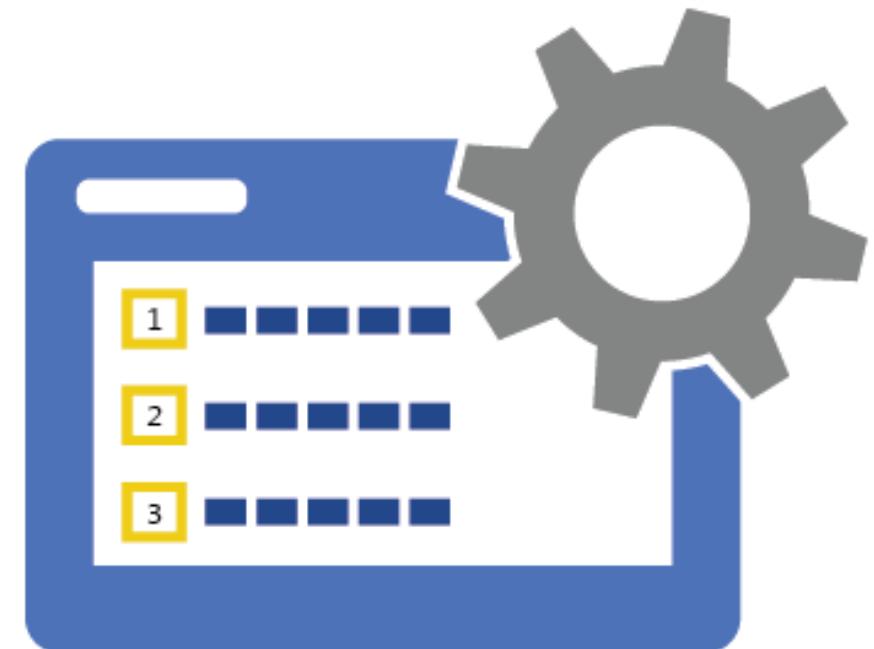
```
def square(num):  
    return num * num
```

You can square this list using the following code:

```
squared_nums = map(square, numbers)
```

Output would be:

```
[1, 4, 9, 16, 25, 36]
```



# Reducer in Python

Reduce written in Python:

Suppose you want to sum the squared numbers:

[1, 4, 9, 16, 25, 36]

Use the **sum** function to add two numbers

```
def sum(a, b ):  
    return a + b
```

You can now sum the numbers using the **reduce** function

```
sum_squared = reduce(sum, squared_nums)
```

Output would be:

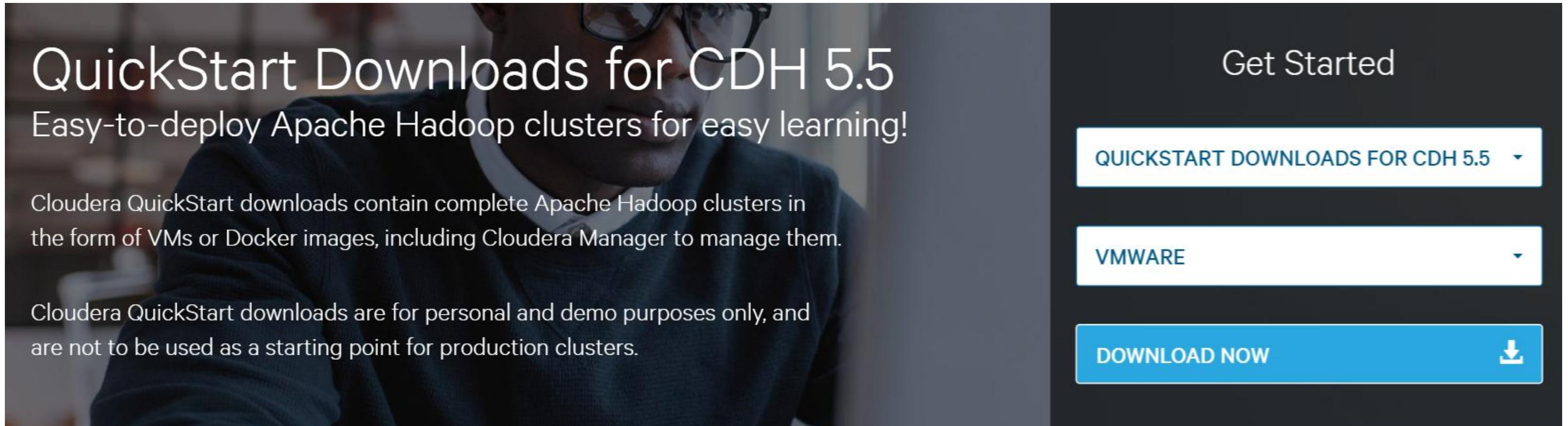
[91]



## Cloudera QuickStart VM Set Up

Cloudera provides enterprise-ready Hadoop Big Data platform which supports Python as well.  
To set up the Cloudera Hadoop environment, visit the Cloudera link:

[http://www.cloudera.com/downloads/quickstart\\_vms/5-7.html](http://www.cloudera.com/downloads/quickstart_vms/5-7.html)



The screenshot shows a landing page for Cloudera QuickStart Downloads. On the left, there's a large image of a person wearing glasses and a suit, looking at a screen. Overlaid on the image is the text "QuickStart Downloads for CDH 5.5" and "Easy-to-deploy Apache Hadoop clusters for easy learning!". Below this, there's a note about the purpose of the downloads. On the right, there's a "Get Started" section with three dropdown menus: "QUICKSTART DOWNLOADS FOR CDH 5.5", "VMWARE", and a blue button labeled "DOWNLOAD NOW" with a download icon.

QuickStart Downloads for CDH 5.5

Easy-to-deploy Apache Hadoop clusters for easy learning!

Cloudera QuickStart downloads contain complete Apache Hadoop clusters in the form of VMs or Docker images, including Cloudera Manager to manage them.

Cloudera QuickStart downloads are for personal and demo purposes only, and are not to be used as a starting point for production clusters.

Get Started

QUICKSTART DOWNLOADS FOR CDH 5.5

VMWARE

DOWNLOAD NOW

Cloudera recommends that you use 7-Zip to extract these files. To download and install it, visit the link:  
<http://www.7-zip.org/>

# Cloudera QuickStart VM: Prerequisites

- These 64-bit VMs require a 64-bit host OS and a virtualization product that can support a 64-bit guest OS.
- To use a VMware VM, you must use a player compatible with WorkStation 8.x or higher:
  - Player 4.x or higher
  - Fusion 4.x or higher
- Older versions of WorkStation can be used to create a new VM using the same virtual disk (VMDK file), but some features in VMware Tools are not available.
- The amount of RAM required varies by the run-time option you choose

CDH and Cloudera Manager Version	RAM Required by VM
CDH 5 (default)	4+ GiB*
Cloudera Express	8+ GiB*
Cloudera Enterprise (trial)	10+ GiB*

# QuickStart VMware Player: Windows, Linux & VMware Fusion: Mac

To launch the VMware, visit the VMware link:

<https://www.vmware.com/products/player/playerpro-evaluation.html>



The screenshot shows the product page for VMware Workstation Player. On the left is a large image of the software's logo, "VMWARE WORKSTATION PLAYER 12". To the right is a blue descriptive box containing text about the software's features, followed by download links for Windows and Linux versions.

**VMware Workstation 12 Player provides a streamlined user interface for creating, running, and evaluating operating systems and applications in a virtual machine regardless of the operating system. With its intuitive interface and virtual machine setup, Workstation Player is the easiest way to deliver a virtual desktop to all of your employees, contractors, or customers. It's now easier than ever to start a trial with VMware Workstation Player.**

**VMware Workstation 12 Player for Windows 64-bit**

**Download Now**

**VMware Workstation 12 Player for Linux 64-bit**

**Download Now**

<https://www.vmware.com/products/fusion/fusion-evaluation.html>



The screenshot shows the product page for VMware Fusion. It features two large images of the software's logo, "VMWARE FUSION 8" and "VMWARE FUSION PRO 8". To the right is a blue descriptive box containing text about the software's features, followed by download links for both versions.

**VMware Fusion 8 is the easiest, fastest and most reliable way to run Windows applications on a Mac without rebooting.**

**VMware Fusion 8 Pro takes virtualization on the Mac to the next level with powerful features designed for advanced users and technical professionals.**

**VMware Fusion 8**

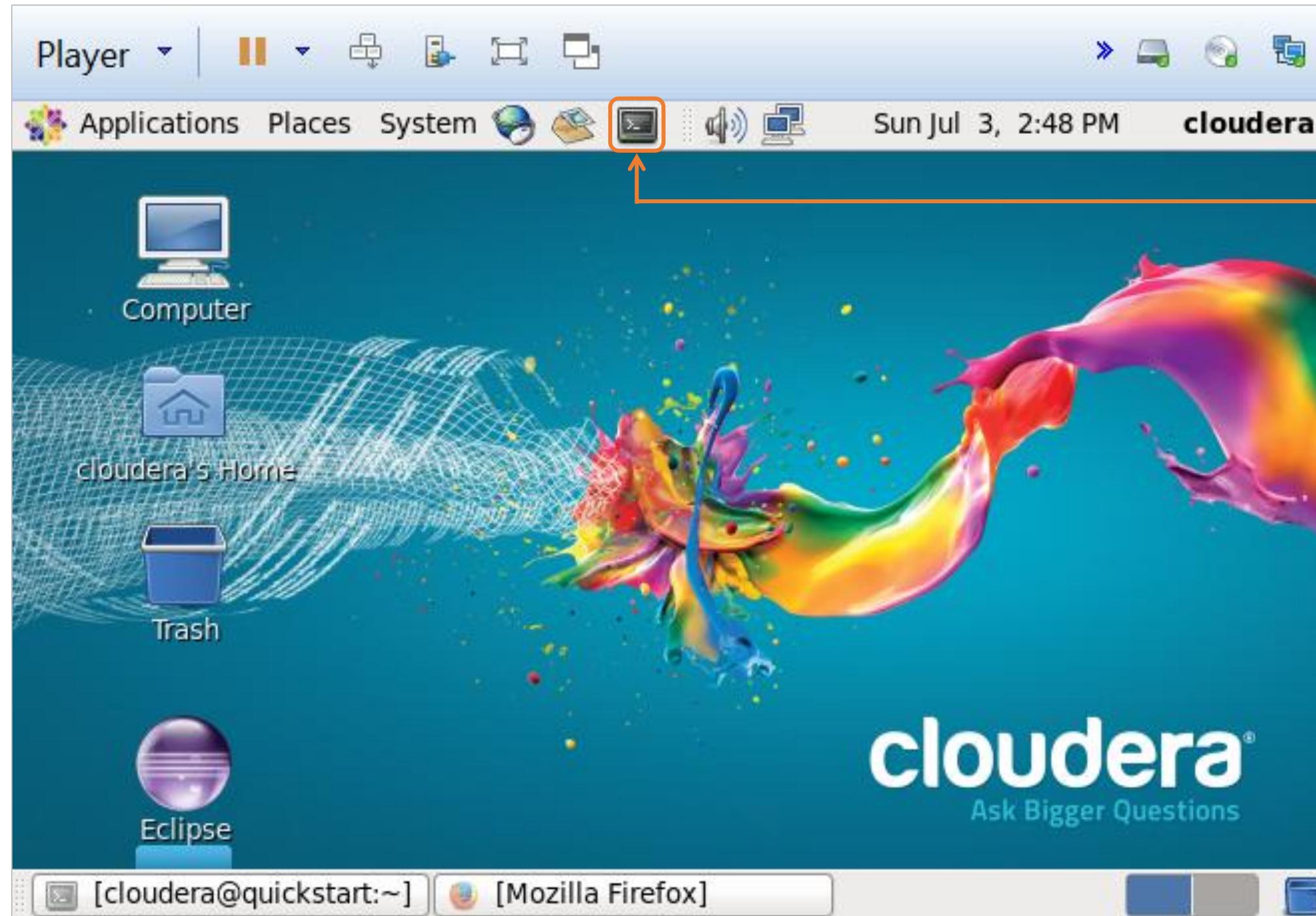
**Download Now**

**VMware Fusion 8 Pro**

**Download Now**

# QuickStart VMware Image

Launch VMware player with Cloudera VM



Launch Terminal

**Account:**  
username: cloudera  
password: cloudera

# QuickStart VM Terminal

Step 01

A screenshot of a terminal window titled "cloudera@quickstart:~". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a command-line interface. The command line shows the prompt "[cloudera@quickstart ~]\$". The terminal is currently empty, displaying only the prompt.

Step 02

A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the user's home directory. The command line shows the prompt "[cloudera@quickstart ~]\$". The user has run the "pwd" command, which outputs the path "/home/cloudera". The user then ran the "ls -lrt" command, which lists all files and directories in the current directory. The output shows a large number of files and directories, including "eclipse", "workspace", "lib", "Documents", "Desktop", "datasets", "cm\_api.sh", "cloudera-manager", "Videos", "Templates", "Public", "Pictures", "Music", "Downloads", "test\_file", "mapper.py", "example\_test\_file", "reducer.py", and "test\_01". The file "cloudera-manager" is highlighted in green. The terminal ends with the prompt "[cloudera@quickstart ~]\$".

Unix command :

- `pwd` to verify present working directory
- `ls -lrt` to list files and directories



## Demo 01—Using Hadoop Streaming for Calculating Word Count

Demonstrate how to create a MapReduce program and use Hadoop Streaming to determine the word count of a document

DATA  
SCIENCE



# Knowledge Check

KNOWLEDGE  
CHECK**What is the usual size of the data block on HDFS?**

- a. 32 MB
- b. 64 MB
- c. 100 MB
- d. 1 GB



KNOWLEDGE  
CHECK**What is the usual size of the data block on HDFS**

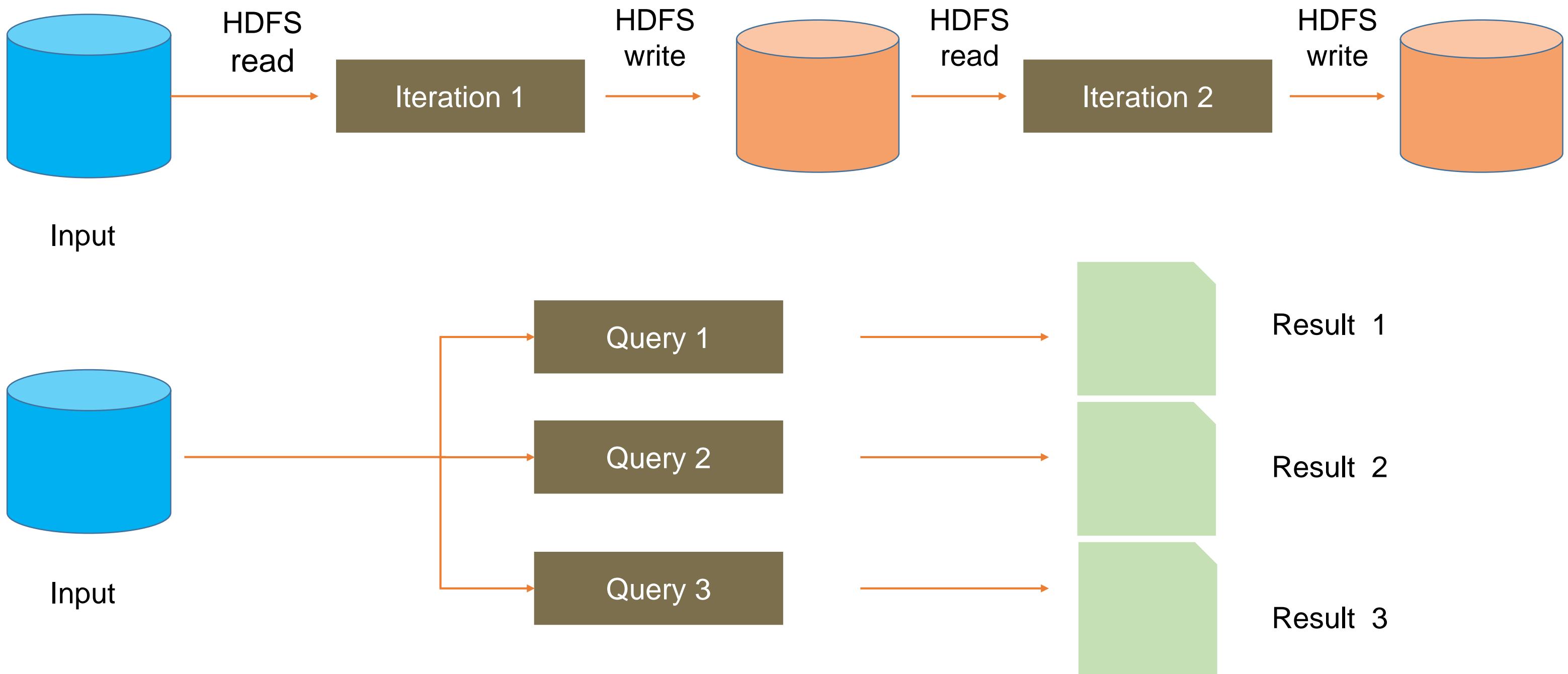
- a. 32 MB
- b. 64 MB
- c. 100 MB
- d. 1 GB



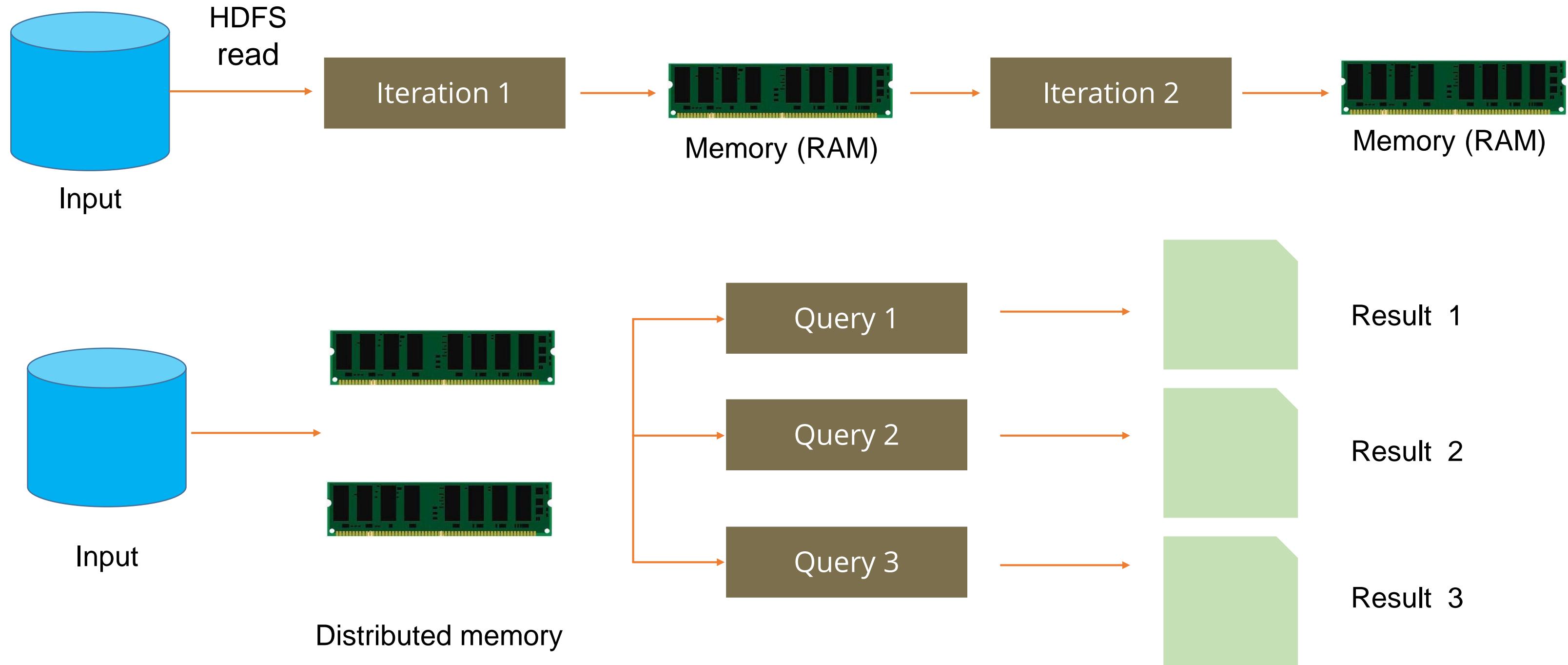
The correct answer is . b.

Explanation The usual data block size on HDFS is 64 MB.

# MapReduce Uses Disk I/O Operations



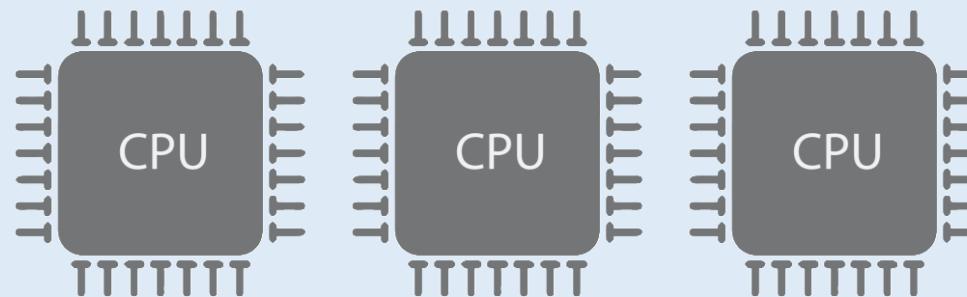
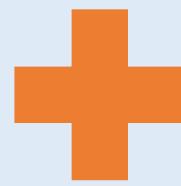
# Apache Spark Uses In-Memory Instead of Disk I/O



10-100 X faster than network and disk

# Hardware Requirements for MapReduce and Spark

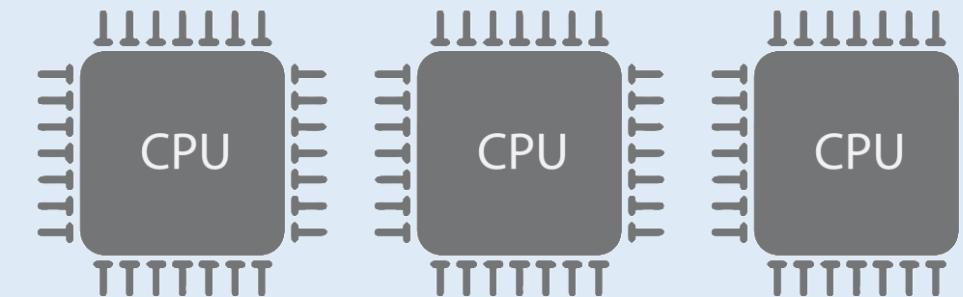
Hard Drives



CPUs

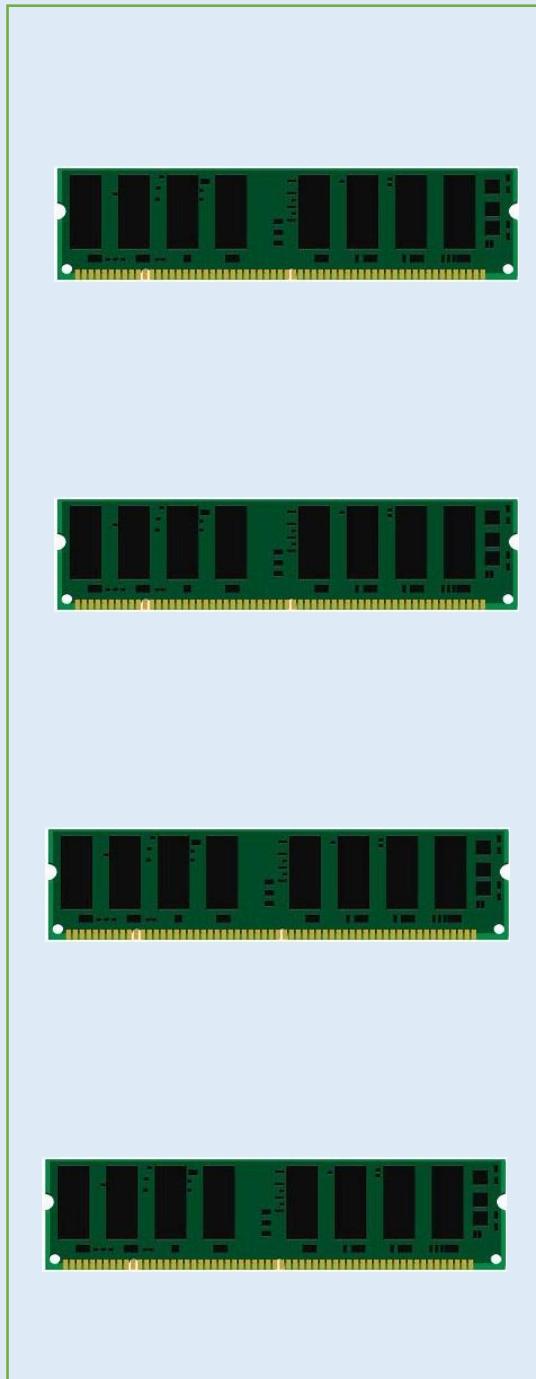
**MapReduce**

Hard Drives



CPUs

**Spark**



Memory

# Apache Spark Resilient Distributed Systems (RDD)

Some basic concepts about Resilient Distributed Datasets (RDD) are listed here:



- The main programming approach of Spark is RDD.
- They are fault-tolerant collections of objects spread across a cluster that you can operate on in parallel. They can automatically recover from machine failure.
- You can create an RDD either by copying the elements from an existing collection or by referencing a dataset stored externally.
- RDDs support two types of operations: transformations and actions.
  - Transformations use an existing dataset to create a new one.
    - Example: Map, filter, join
  - Actions compute on the dataset and return the value to the driver program.
    - Example: Reduce, count, collect, save



If the available memory is insufficient, then the data is written to disk.

# Advantages of Spark



Listed here are some of the advantages of using Spark:

**Faster:**

10 to 100 times faster than Hadoop MapReduce

**Simplified:**

- Simple data processing framework
- Interactive APIs for Python for faster application development

**Efficient:**

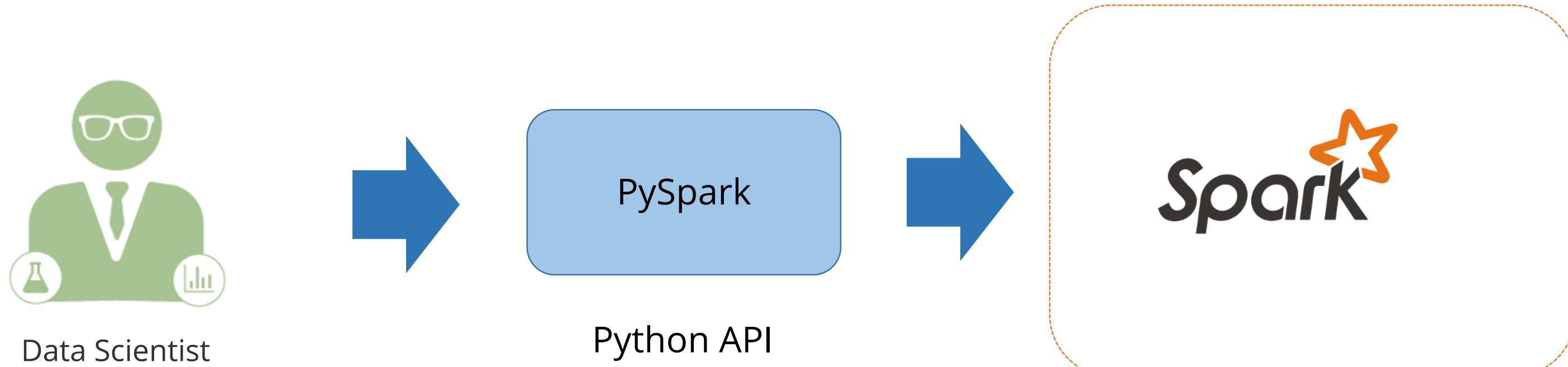
Has multiple tools for complex analytics operations

**Integrated:**

Can be easily integrated with existing Hadoop infrastructure

# PySpark : Python API for Spark

PySpark is the Spark Python API which enables data scientists to access Spark programming model



# PySpark : RDD Transformations and Actions

## Transformation

Transformation	Description
map()	Returns RDD, formed by passing data element of the source
filter()	Returns RDD based on selection
flatMap()	Maps items present in the dataset and returns sequence
reduceByKey()	Returns key value pairs where values for which each key is aggregated by value

## Action

Action	Description
collect()	Returns all elements of the dataset as an array
count()	Returns the number of elements present in the dataset
first()	Returns the first element in the dataset
take(n)	Returns number of elements (n) as specified by the number in the parenthesis

SparkContext or SC is the entry point to spark for the spark application

# Spark Tools

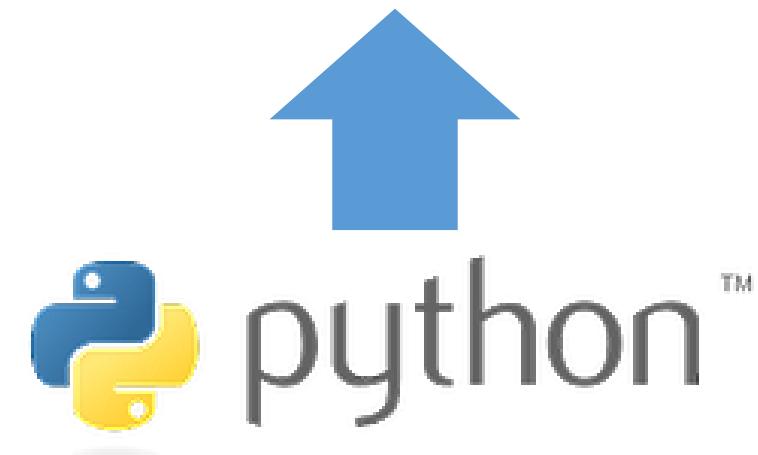
Spark  
SQL

Spark  
Streaming

MLlib  
(machine  
Learning)

GraphX  
(graph)

Spark



Interactive Python APIs

# Apache Spark Set Up

To set up the Apache Spark environment, access the link:

<http://spark.apache.org/downloads.html>

Please use [7-Zip](#) to extract these files.



The screenshot shows the official Apache Spark website. At the top is the Apache Spark logo, which includes a stylized orange star above the word "Spark". Below the logo is the tagline "Lightning-fast cluster computing". A blue navigation bar spans the width of the page, containing links for "Download", "Libraries", "Documentation", "Examples", "Community", and "FAQ".

## Download Apache Spark™

Our latest stable version is Apache Spark 1.6.2, released on June 25, 2016 ([release notes](#)) ([git tag](#))

1. Choose a Spark release:
2. Choose a package type:
3. Choose a download type:
4. Download Spark: [spark-1.6.2-bin-hadoop2.4.tgz](#)
5. Verify this release using the [1.6.2 signatures and checksums](#).

Note: Scala 2.11 users should download the Spark source package and build [with Scala 2.11 support](#).

# Apache Spark : Environment Variable Set Up

Environment Variables

The screenshot shows the Windows Environment Variables dialog box. It has two main sections: 'User variables for niteen' and 'System variables'. In the 'User variables for niteen' section, the 'SPARK\_HOME' variable is highlighted with an orange border. In the 'System variables' section, the 'Path' variable is highlighted with a blue border. At the bottom of each section are 'New...', 'Edit...', and 'Delete...' buttons. Below the sections are 'OK' and 'Cancel' buttons.

Variable	Value
PATH	C:\Niteen\Anaconda2;C:\Niteen\Anaconda2\Scripts;C:\Niteen\An...
SPARK_HOME	C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Value
MONETDB_INSTALL_DIR	C:\Pentaho\monetdb
NUMBER_OF_PROCESSORS	4
OnlineServices	Online Services
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;C:\Program Files (x86)\Inte...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PENTAHO_HOME	C:\Pentaho
PFNTAHO_INSTALLED_ICF...	C:\Pentaho\installed\licenses.xml

[installed directory]\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4

[installed directory] \spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin

# Apache Spark: Jupyter Notebook Integration

Command Prompt - pyspark

```
C:\Users\niteen>cd C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin  
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>set PYSPARK_DRIVER_PYTHON=ipython  
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>set PYSPARK_DRIVER_PYTHON_OPTS=notebook  
C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin>pyspark  
[W 19:33:44.451 NotebookApp] Permission to listen on port 8888 denied  
[I 19:33:44.612 NotebookApp] Serving notebooks from local directory: C:\NITEEN\software\spark-1.6.1-bin-hadoop2.4\spark-1.6.1-bin-hadoop2.4\bin  
[I 19:33:44.618 NotebookApp] 0 active kernels  
[I 19:33:44.618 NotebookApp] The Jupyter Notebook is running at: http://localhost:8889/  
[I 19:33:44.716 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Setup the pyspark notebook specific variables

jupyter PySpark - Env Set Up Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Help

CellToolbar

```
In [1]: sc
```

Out[1]: <pyspark.context.SparkContext at 0x48b9a20>

```
In [ ]:
```

Run the pyspark command

Check SparkContext



## Demo 02—Using PySpark to Determine Word Count

Demonstrate how to use the Jupyter integrated PySpark API to determine the word count of a given dataset

DATA  
SCIENCE



# Knowledge Check

KNOWLEDGE  
CHECK**What happens if the available memory is insufficient while performing RDD transformations?**

- a. The RDD process waits for memory to be available
- b. The process is cancelled by scheduler
- c. The data is written to the disk
- d. The RDD process fails



KNOWLEDGE  
CHECK**What happens if the available memory is insufficient while performing RDD transformations?**

- a. The RDD process waits for memory to be available
- b. The process is cancelled by scheduler
- c. The data is written to the disk
- d. The RDD process fails



The correct answer is . c.

Explanation The data is written to the disk in case the memory is insufficient while performing transformations.



Problem

Instruction  
s

To determine the word count of the given Amazon dataset:

- Create a MapReduce program to determine the word count of the Amazon dataset
- Submit the MapReduce task to HDFS and run it
- Verify the output

*Click each tab to know more. Click the Resources tab to download the files for this assignment.*

Problem

Instruction  
s

Instructions on performing the assignment:

- Download the “Amazon text dataset.txt” file from the “Resource” tab. Use the QuickStart VM terminal to create a file and copy-paste the Amazon dataset into it.

Special instructions:

- This assignment is done purely on Cloudera’s QuickStart VM. You may need to learn a few basic UNIX commands to operate the program.
- For any cues, refer the Hadoop Streaming demo provided in the lesson.



## Problem

## Instructions

Use the given dataset to count and display all the airports based in New York using PySpark. Perform the following steps:

- View all the airports listed in the dataset
- View only the first 10 records
- Filter the data for all airports located in New York
- Clean up the dataset, if required

Problem

Instruction  
s

Instructions on performing the assignment:

- Download the “Airport.csv” file from the “Resource” tab. You can load the saved file to the Jupyter notebook that you would be using to complete the assignment..

Common instructions:

- If you are new to Python, download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps for installing Anaconda and the Jupyter notebook.
- Download the “Assignment 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the assignment.



**QUIZ**

1

**What are the core components of Hadoop? *Select all that apply.***

- a. MapReduce
- b. HDFS
- c. Spark
- d. RDD



**QUIZ**  
**1**

**What are the core components of Hadoop? Select all that apply.**

- a. MapReduce
- b. HDFS
- c. Spark
- d. RDD



The correct answer is

. **a & b**

**Explanation:** MapReduce and HDFS are the core components of Hadoop.

**QUIZ**  
**2**

**MapReduce is a data processing framework which gets executed \_\_\_\_.**

- a. at DataNode
- b. at NameNode
- c. on client side
- d. in memory



**QUIZ  
2**

**MapReduce is a data processing framework which gets executed \_\_\_\_.**

- a. at DataNode
- b. at NameNode
- c. on client side
- d. in memory



The correct answer is . **a**

**Explanation:** The MapReduce program is executed at the data node and the output is written to the disk.

**QUIZ  
3**

**Which of the following functions is responsible for consolidating the results produced by each of the Map() functions/tasks?**

- a. Reducer
- b. Mapper
- c. Partitioner
- d. All of the above



**QUIZ  
3**

**Which of the following functions is responsible for consolidating the results produced by each of the Map() functions/tasks?**

- a. Reducer
- b. Mapper
- c. Partitioner
- d. All of the above



The correct answer is . **a**

**Explanation:** Reducer combines or aggregates results produced by mappers.

**QUIZ**

4

**What transforms input key-value pairs to a set of intermediate key-value pairs?**

- a. Mapper
- b. Reducer
- c. Combiner
- d. Partitioner



**QUIZ**

4

**What transforms input key-value pairs to a set of intermediate key-value pairs?**

- a. Mapper
- b. Reducer
- c. Combiner
- d. Partitioner

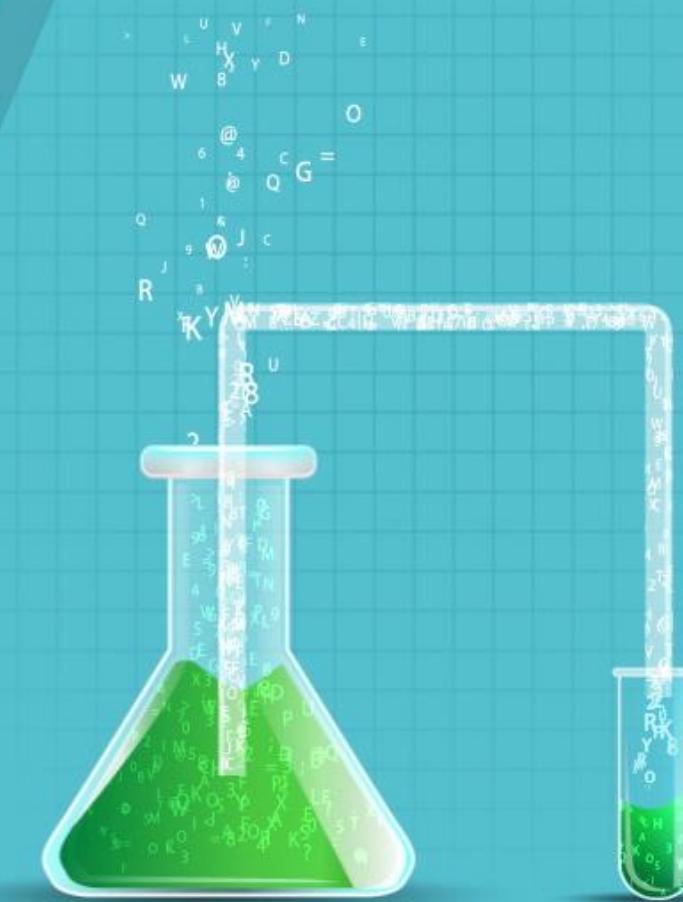


The correct answer is . **a**

**Explanation:** Mapper processes input data to intermediate key-value pairs which are in turn processed by reducers.

# Key Takeaways

- As Python is a Data Scientist's preferred choice of language, it is important to provide Big Data solutions that accommodates it.
- There are two primary components of Hadoop architecture: Hadoop Distributed File System or HDFS and MapReduce.
- Both Hadoop and Spark provide Python APIs to help Data Scientists use the Big Data platform.
- MapReduce has two functions—mapper and reducer.
- MapReduce carries out computations on data through disk I/O operations while Apache Spark carries them out in-memory.
- The main programming approach of Spark is RDD.
- Spark is almost 10 to 100 times faster than Hadoop MapReduce.
- There are mainly four components in Spark tools: Spark SQL, Spark Streaming, Mllib, and GraphX.



**This concludes “Python Integration with MapReduce and Spark”.**

This is the final lesson of the Data Science with Python Course.



After learning about Data Science in depth, it is now time to implement the knowledge gained through this course in real-life scenarios. We will provide you with four scenarios where you need to implement data science solutions. To perform these tasks, you can use the different Python libraries such as NumPy, SciPy, Pandas, scikit-learn, matplotlib, BeautifulSoup, and so on. You will focus on acquiring stock data information for the companies listed.

The scope of the project is as follows:

Problem

Instructions

Solution

- Import the financial data using Yahoo data reader for the following companies:
  - Yahoo
  - Apple
  - Amazon
  - Microsoft
  - Google
- Perform fundamental data analysis
  - Fetch the last one year's data
  - View the values of Apple's stock
  - Display the plot of closing price
  - Display the stock trade by volume
  - Plot all companies' data together for closing prices

Problem

Instructions

Solution

- Perform Daily Return Analysis and show the relationship between different stocks
  - Plot the percentage change plot for Apple's stock
  - Show a joint plot for Apple and Google
  - Use PairPlot to show the correlation between all the stocks
- Perform risk analysis

Problem

Instructions

Solution

Instructions to perform the project:

- Download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps to install Anaconda and the Jupyter notebook.
- Download the “Project 01” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the project.

We recommend you to first solve the project and then view the solution to assess your learning.

Problem

Instructions

Solution

Hope you had a good experience working on the project "Stock Market Data Analysis."  
Go to the next screen to assess your performance.

Click **Next** to view the demo.



After learning about Data Science in depth, it is time to implement the knowledge gained through this course in real-life scenarios. We are providing four real-life scenarios where you can implement data science solutions. To develop solutions to these problems, you can use various Python libraries like NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, BeautifulSoup, and so on.  
Project details are given below:

Problem

Instructions

Solution

## Titanic Dataset Analysis

On April 15, 1912, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This tragedy shocked the world and led to better safety regulations for ships. Here, we ask you to perform the analysis through the exploratory data analysis technique. In particular, we want you to apply the tools of machine learning to predict which passengers survived the tragedy.

The details of these projects and their scope are listed in the following sections.

*Click each tab to know more. Click the Resources tab to download the files for this project.*

Problem

Instructions

Solution

## **Titanic Dataset Analysis**

- Data acquisition of the Titanic dataset
  - train dataset
  - test dataset
- Perform the Exploratory Data Analysis (EDA) - for train dataset
  - passengers age distribution
  - passengers survival by age
  - passengers survival breakdown
  - passengers class distribution
  - passengers embarkation by locations

*Click each tab to know more. Click the Resources tab to download the files for this project.*

Problem

Instructions

Solution

## Titanic Dataset Analysis

- Perform machine learning to train the machine model and
  - create user defined function to load train data set
  - create user defined function to load test data set
  - create machine model
  - train the machine
  - predict whether a passenger survived the tragedy or not
  - persist the mode for future re-use
  -

*Click each tab to know more. Click the Resources tab to download the files for this project.*

Problem

Instructions

Solution

Instructions to perform the project:

- Download the “Anaconda Installation Instructions” document from the “Resources” tab to view the steps to install Anaconda and the Jupyter notebook.
- Download the “Project 02” notebook and upload it on the Jupyter notebook to access it.
- Follow the provided cues to complete the project.

We recommend you to first solve the project and then view the solution to assess your learning.

Problem

Instructions

Solution

Hope you had a good experience working on the project “Titanic data set analysis.”

Go to the next screen to assess your performance.

Click **Next** to view the demo.

*To view the demo for this project, click Next.*

Thank You