Unit II

Introduction to Analytics, Introduction to Tools and Environment, Application of Modeling in Business, Databases & Types of Data and variables, Data Modeling Techniques Missing Imputations etc. Need for Business Modeling.

Data Analytics:

- Data Analytics refers to the techniques used to analyze current data/historical data of an organization to enhance productivity and business gain and for better decision making.

or

- Analytics is a journey that involves a combination of potential skills, advanced technologies, applications, and processes used by firm to gain business insights from data and statistics. This is done to perform business planning.

or

- Data analytics is the process of collecting, cleaning, and analyzing data to extract meaningful insights, It involves using statistical methods, machine learning algorithms, and data visualization techniques. It is used by businesses of all sizes to make informed decisions, improve efficiency, and gain a competitive advantage.

Some key reasons for the need of analytics include:

- **Informed Decision-Making:** Analytics helps to make smart decisions by relying on facts and data.
- **Efficiency Improvement:** By analysing data, organizations can identify areas for improvement, streamline processes, and enhance overall operational efficiency.
- **Reduced costs:** Data analytics can help businesses identify and eliminate unnecessary costs.
- **Improved customer satisfaction:** Data analytics can help businesses understand their customers better and develop products and services that meet their needs

Applications of Data Analytics:

- Business Intelligence
- Healthcare

- Finance
- Government
- Education
- Manufacturing
- Marketing and Sales

Business Intelligence (BI): Data analytics helps businesses make informed decisions by analyzing and interpreting data related to their operations, customers, and market trends. Ex: A retail company analyzes sales data to understand which products are popular among customers, helping them optimize inventory and marketing strategies.

Healthcare: In healthcare, data analytics can be used to improve patient care, optimize hospital operations, identify trends in diseases and drug discovery process.

Finance: Finance uses data analytics to manage risk, detect fraud, and optimize investment portfolios. For example, banks use data analytics to identify customers who are at risk of defaulting on their loans.

Government: Data analytics helps governments make better decisions by analyzing data from various sources. This leads to more effective policymaking & improved public services.

Manufacturing: Data analytics helps factories run smoother by spotting problems early, optimizing production, and streamlining deliveries.

Marketing and Sales: Data analytics empowers marketing teams to understand customer preferences, segment markets effectively, and target campaigns strategically. It drives personalized marketing campaigns, increases customer engagement, and boosts sales.

Types of Analytics:

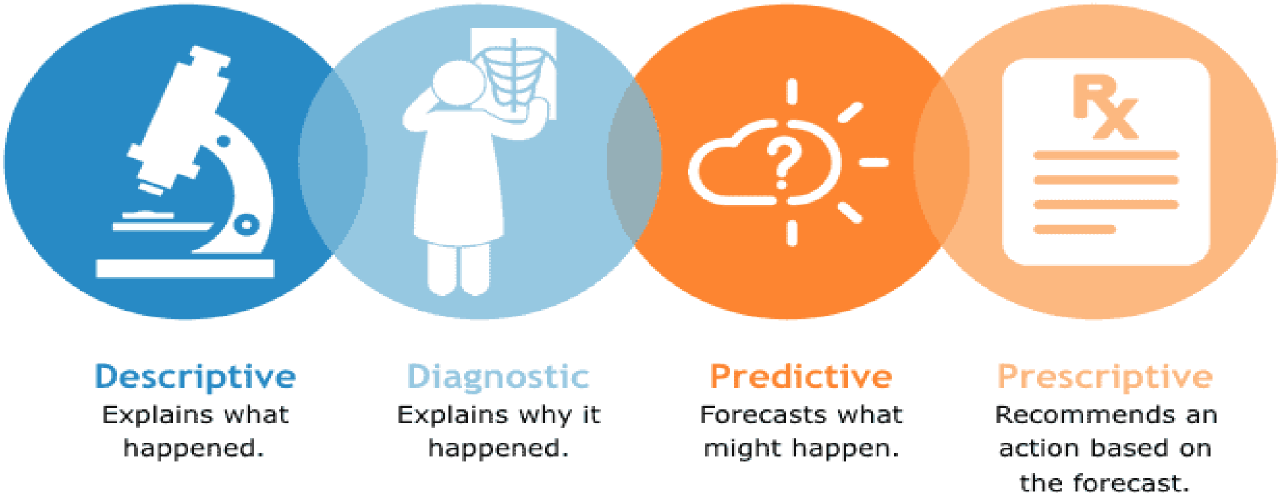
For an organization huge amount of data is generated at various steps. Depending on the situation their might be requirement of data analysis, there are four main kinds of analytics

- Descriptive Analytics
- Diagnostic Analytics
- Predictive Analytics
- Prescriptive Analytics.

These four types together answer everything a company needs to know:

"from what's going on in the company to what solutions to be adopted"

The four types of analytics are usually implemented in stages and no one type of analytics is said to be better than the other. They are interrelated and each of these offers a different insight.


Rx
Descriptive
Explains what happened.
Diagnostic
Explains why it happened.
Predictive
Forecasts what might happen.
Prescriptive
Recommends an action based on the forecast.

Descriptive Analytics: Descriptive analytics is used to summarize and describe historical data. It can be used to answer questions such as: *"What happened?" (or When did it happen?).*

Descriptive analytics is typically used to create reports and charts that can be used to track trends, identify patterns, and make comparisons. For example, Data analysts who are working with an e-commerce marketing team to review sales data will identify the sales trends and patterns, we will see an increase or decrease in sales from last year, specifically in what region and by what percentage.

Diagnostic analytics: It is used to identify the root causes of problems. It can be used to answer questions such as: **Why did something happen? Or what factors contributed to the problem? Or what can be done to prevent the problem from happening again?**

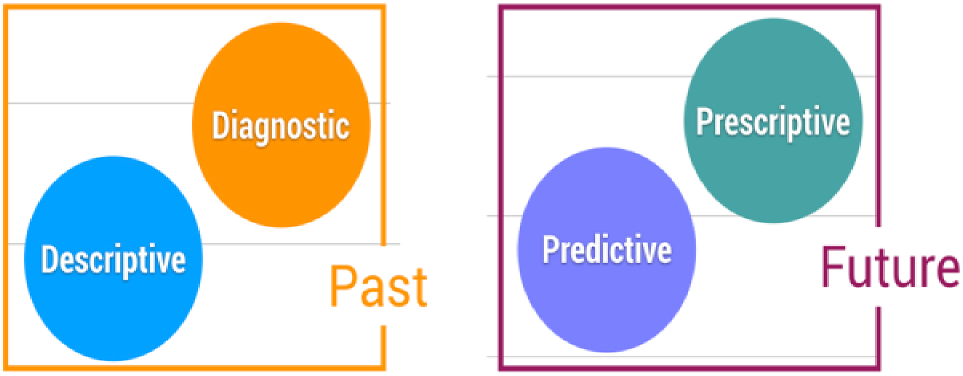
Diagnostic analytics is typically used to drill down into historical data to identify patterns and anomalies. This information can then be used to develop solutions to problems. Ex:, from descriptive analytics Data Analyst will understand the increase or decrease in sales. If drop in sales means we have to identify the reason or root of cause, why there is a drop in sales.

Predictive Analytics: Predictive Analytics is another type of advanced analytics that looks to use data and information to answer the question **"What is likely to happen?".**

Predictive analytics is a type of data analytics that uses statistical models and machine learning techniques to predict future outcomes. It can be used to forecast trends, identify risks, and make better decisions. From Descriptive and Diagnostic we are able to identify the reason for the drop in sales. Predictive analytics helps to find what would be the expected sale in the next month, quarter, or year, etc.

Prescriptive Analytics: Prescriptive Analytics is a method of analytics that analyzes data to answer the question **"What should be done?"** Prescriptive analytics is a type of data analytics that uses historical data, predictive analytics, and machine learning to recommend the best course of action to take in a given situation.

Prescriptive analytics is comparatively complex in nature and many companies are not yet using them in day-to-day business activities, as it becomes difficult to manage. Prescriptive analytics can help seller to setup the best prices for new products by analyzing various factors like cost, competition, and demand. This helps retailers maximize profits and avoid costly pricing mistakes.



"D"s are about the past and the "P"s are in the future

Introduction to Tools and Environment

- Data analytics is the process of collecting, cleaning, and analyzing data to extract meaningful insights, It involves using statistical methods, machine learning algorithms, and data visualization techniques. It is used by businesses of all sizes to make informed decisions, improve efficiency, and gain a competitive advantage.
- In the field of data analytics, various tools and environments play a crucial role in managing and analyzing data effectively. Here's an introduction to some key aspects of tools and environments in data analytics(Some are programming based and others are non-programming based.):
    - ***Microsoft Excel***
    - ***Python***
    - ***R Programming***
    - ***Jupyter Notebook***
    - ***Apache Spark***
    - ***SAS***
    - ***Microsoft Power BI***
    - ***Tableau***
    - ***KNIME***

Python: Python is a free and easy-to-learn programming language that can be used for many things, like collecting, analyzing, and reporting data. It has lots of useful tools, but it needs more computer memory than other languages, so it might not be as fast.

- **Type of tool:** Programming language.
- **Availability:** Open-source, with thousands of free libraries.
- **Used for:** Everything from data scraping to analysis and reporting.
- **Pros:** Easy to learn, highly versatile, widely-used.
- **Cons:** Memory intensive—doesn’t execute as fast as some other languages.

R Programming: R is a free programming language that is good for statistics and data mining, but it's not as fast or easy to use as Python. R can work on different computers and has many helpful tools.

- **Type of tool:** Programming language.
- **Availability:** Open-source.

- **Mostly used for:** Statistical analysis and data mining.
- **Pros:** Platform independent, highly compatible, lots of packages.
- **Cons:** Slower, less secure, and more complex to learn than Python.

Jupyter Notebook : Jupyter Notebook is a free and easy-to-use tool that is great for analyzing large amounts of data. It is fast and user-friendly, but it does not have built-in file management and its interface can be a bit rigid.

- **Type of tool:** Data processing framework.
- **Availability:** Open-source.
- **Mostly used for:** Big data processing, machine learning.
- **Pros:** Fast, dynamic, easy to use.
- **Cons:** No file management system, rigid user interface.

Microsoft Power BI is a tool that helps businesses see their data and make predictions. It can be used to create charts and graphs, and to find trends and patterns in data. This information can then be used to make better business decisions.

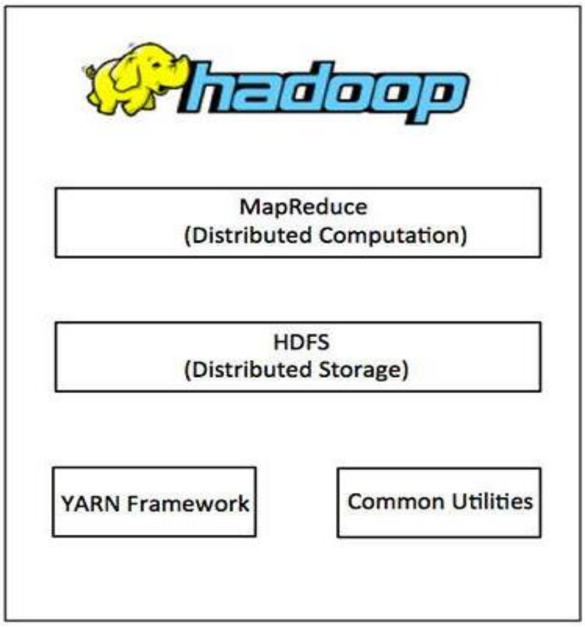
- **Type of tool:** Business analytics suite.
- **Availability:** Commercial software (with a free version available).
- **Mostly used for:** Everything from data visualization to predictive analytics.
- **Pros:** Great data connectivity, regular updates, good visualizations.
- **Cons:** Clunky user interface, rigid formulas, data limits (in the free version).

Tableau: A commercial data visualization tool is good for making charts and graphs, but it's missing some important features for managing and preparing data.

- **Type of tool:** Data visualization tool.
- **Availability:** Commercial.
- **Mostly used for:** Creating data dashboards and worksheets.
- **Pros:** Great visualizations, speed, interactivity, mobile support.
- **Cons:** Poor version control, no data pre-processing.

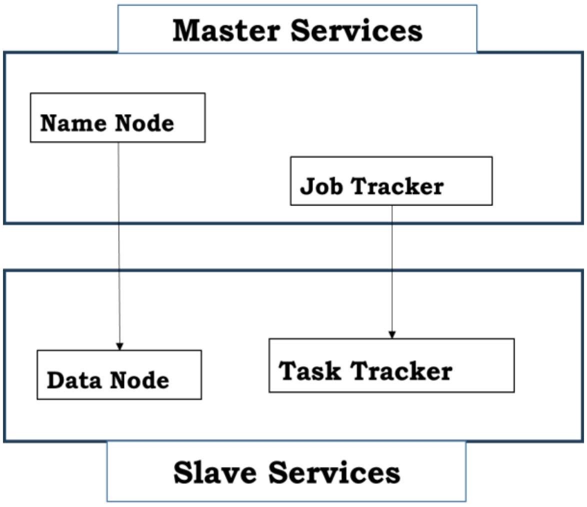
Hadoop: Hadoop, an open-source framework by Apache, manages large volumes of data through storage, processing, and analysis. Primarily written in Java, it focuses on batch/offline processing. Major tech companies like Facebook, Yahoo, Google, Twitter, and LinkedIn use Hadoop, which offers scalability by adding nodes to the cluster.

Modules of Hadoop


hadoop
MapReduce (Distributed Computation)
HDFS (Distributed Storage)
YARN Framework
Common Utilities

- **Hadoop Common:** Foundation tools and utilities shared by other Hadoop modules.
- **Hadoop Distributed** File System (HDFS): Manages storage, storing data across a distributed network.
- **Hadoop YARN((Yet Another Resource Negotiator):** Handles resource management, ensuring efficient use of cluster resources.
- **Hadoop MapReduce:** Processes and analyzes large-scale data in a distributed environment.

HDFS Services:



- Name Node
- Job Tracker
- Data Node
- Task Tracker

Every master service can communicate with each other

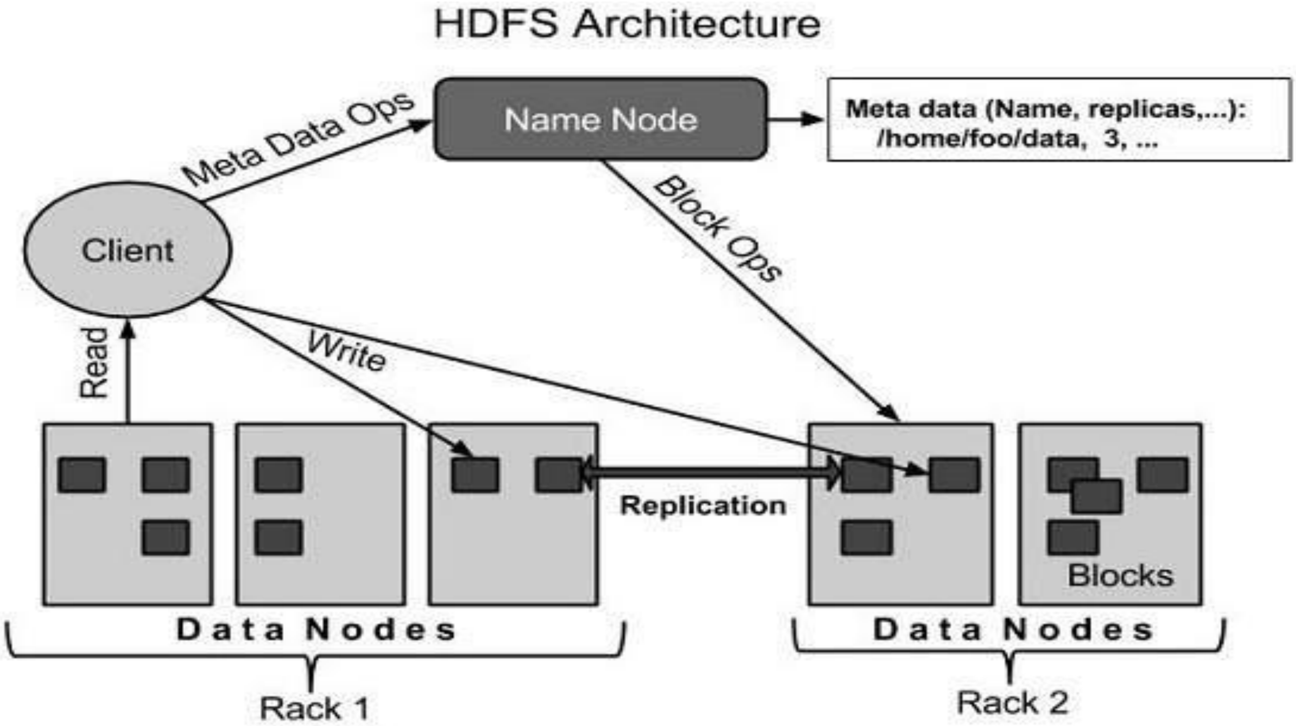
Every Slave service can communicate with each other.

Name Node: The NameNode in Hadoop is like the central command center for the file system. It manages file details, coordinates with DataNodes for data storage, and ensures the reliability and integrity of the entire distributed system. In essence, it's the vital hub that keeps everything organized and running smoothly.

Data Node: The DataNode in Hadoop is like a storage worker. Its main job is to keep data safe by storing it in blocks, making copies for backup, and checking with the boss (NameNode) regularly to ensure everything is in good shape. It's a key player in making data is safe and available when needed in a Hadoop cluster.

Job Tracker & Task Tracker:

In the Hadoop MapReduce framework, the Job Tracker coordinates job execution by breaking tasks into smaller units, assigning them to Task Trackers, and overseeing their progress. It manages task re-execution in case of failures and maintains an overall job status. Task Trackers execute assigned Map and Reduce tasks, regularly reporting their status to the Job Tracker. Together, they ensure efficient and fault-tolerant processing of large-scale data in a Hadoop cluster.



Map Reduce:

MapReduce is a programming paradigm that runs in the background of Hadoop to provide scalability and easy data-processing solutions.

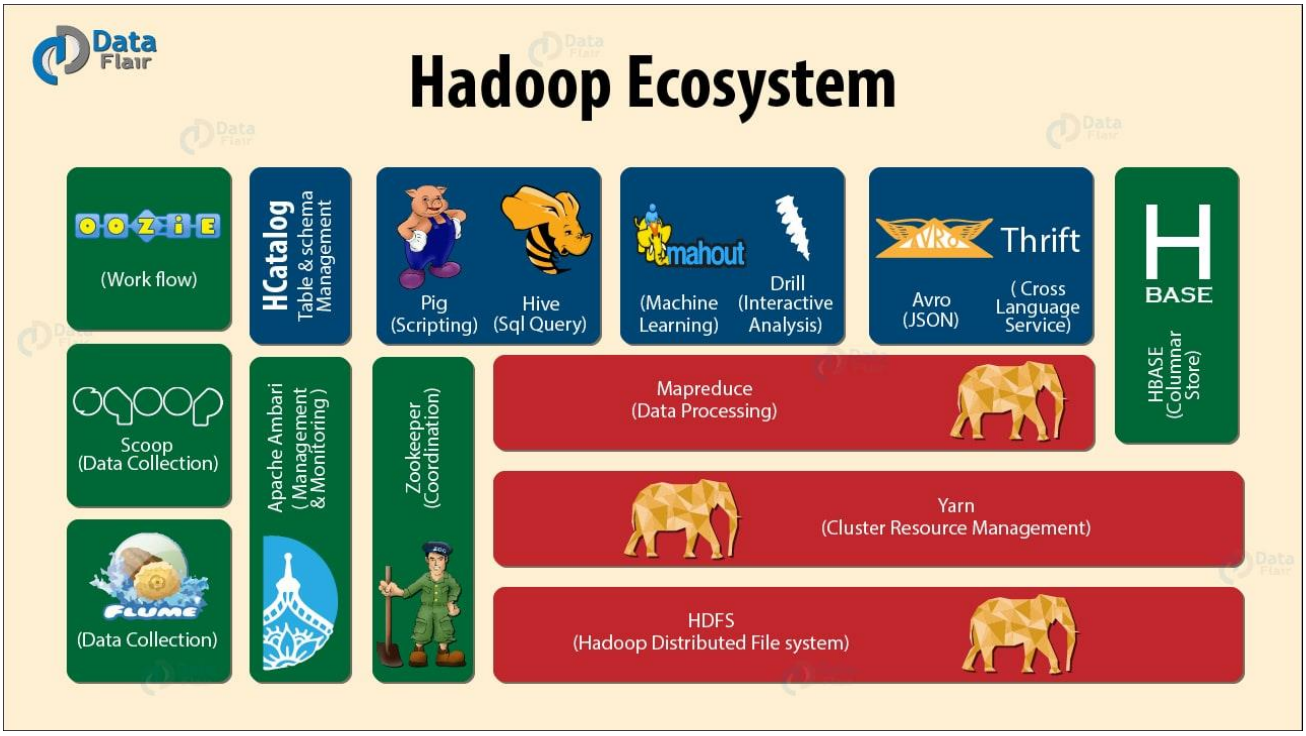
The MapReduce algorithm contains two important tasks, namely Map and Reduce.

- ✓ The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key-value pairs).
- ✓ The Reduce task takes the output from the Map as an input and combines those data tuples (key-value pairs) into a smaller set of tuples.

Hadoop Ecosystem: Beyond ***HDFS, YARN and MapReduce,*** the entire Apache Hadoop "platform" is now commonly considered to consist of a number of related projects as well:

- ✓ Apache Pig
- ✓ Apache Hive

- ✓ Apache Hbase
- ✓ Apache Ambari
- ✓ Zoo Keeper
- ✓ Sqoop
- ✓ Oozie
- ✓ Mahout (Machine Learning) and others.



Hive: Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Features of Hive

- ✓ It stores schema in a database and processed data into HDFS.
- ✓ It provides SQL type language for querying called HiveQL or HQL.
- ✓ It is familiar, fast, scalable, and extensible.

PIG: This is a high-level scripting language used to execute queries for larger datasets that are used within Hadoop. Pig's simple SQL-like scripting language is known as Pig Latin and its main objective is to perform the required operations and arrange the final output in the desired format.

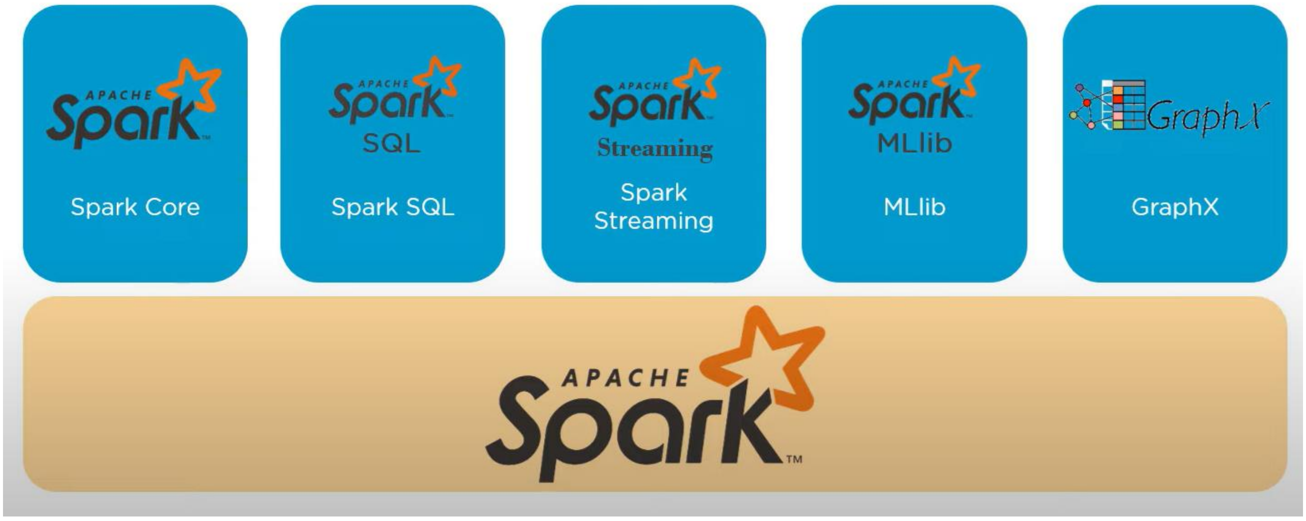
Features of Pig:

- ✓ Rich set of operators

- ✓ Ease of Programming
- ✓ Externsibility
- ✓ Handles all kinds of Data.

Apache Spark:

- Apache Spark is an open-source data processing engine to store and process data in real-time across various clusters of computers using simple programming constructs. Support various programming languages such as R, Python, Java, Scala. Developers and data scientists incorporate Spark into their applications to rapidly query, analyze, and transform data at scale
- **Fast processing:** Spark contains Resilient Distributed Datasets (RDD) which saves time taken in reading, and writing operations and hence, it runs almost ten to hundred times faster than Hadoop.
- **In-memory computing:** In Spark, data is stored in the RAM, so it can access the data quickly and accelerate the speed of analytics
- **Flexible:** Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python.
- **Fault tolerance:** Spark contains Resilient Distributed Datasets (RDD) that are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero
- **Better analytics:** Spark has a rich set of SQL queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed better.



Spark Core: Spark Core is the base engine for large-scale parallel and distributed data processing. It is responsible for: memory management, scheduling, distributing and monitoring jobs on a cluster, fault recovery, interacting with storage systems.

Spark SQL: The Spark SQL is built on the top of Spark Core. It provides support for structured data.

- It allows to query the data via SQL (Structured Query Language) as well as the Apache Hive variant of SQL called the HQL (Hive Query Language).
- It supports JDBC and ODBC connections that establish a relation between Java objects and existing databases, data warehouses and business intelligence tools.
- It also supports various sources of data like Hive tables, Parquet, and JSON.

GraphX: The GraphX is a library that is used to manipulate graphs and perform graph-parallel computations.

Scala is a high-level programming language which is a combination of object-oriented programming and functional programming. t is a preferred language for big data processing frameworks like Apache Spark due to its performance, scalability, and compatibility with Java.

Features of Scala:

- ✓ Type Inference
- ✓ Immutability
- ✓ Case classes and Pattern matching
- ✓ Rich Set of Collection

Impala: Impala is crucial for real-time, interactive SQL queries on large-scale data stored in Hadoop Distributed File System (HDFS). As a massively parallel processing (MPP) query engine, Impala enables fast analytics, making it essential for businesses requiring quick insights from big data without the need for time-consuming batch processing**.**

Features of Impala

- ✓ Real-Time Query Processing
- ✓ Massively Parallel Processing (MPP)
- ✓ Compatibility with Apache Hive
- ✓ Support for Hadoop File Formats

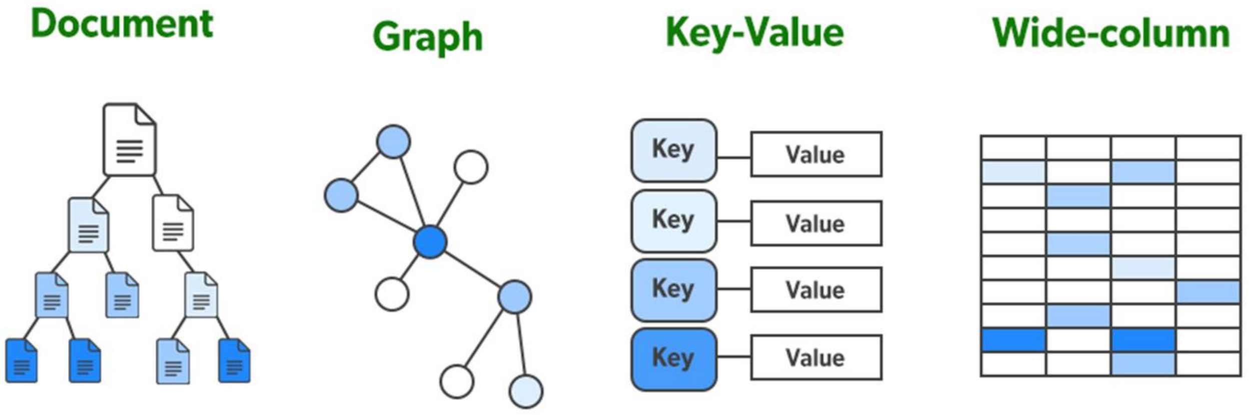
Cluster computing: Cluster computing is teamwork for computers, where multiple machines work together to solve big problems faster through parallel processing. It's scalable, fault-tolerant, and widely used for tasks like scientific simulations and data analysis.

Databases & Types of Data and variables

- **Data Base:** A Database is a collection of related data.
- **Database Management System:** DBMS is a software or set of Programs used to define, construct and manipulate the data.
- **Relational Database Management System:** A RDBMS, or relational database management system, is a type of database that stores data in tables and uses relationships between those tables to organize and manage the data. A RDBMS stores data in tables, which are made up of rows and columns. Each row represents a record, and each column represents an attribute of that record. For example, a table of customer data might have rows for each customer and columns for the customer's name, address, phone number, and email address.

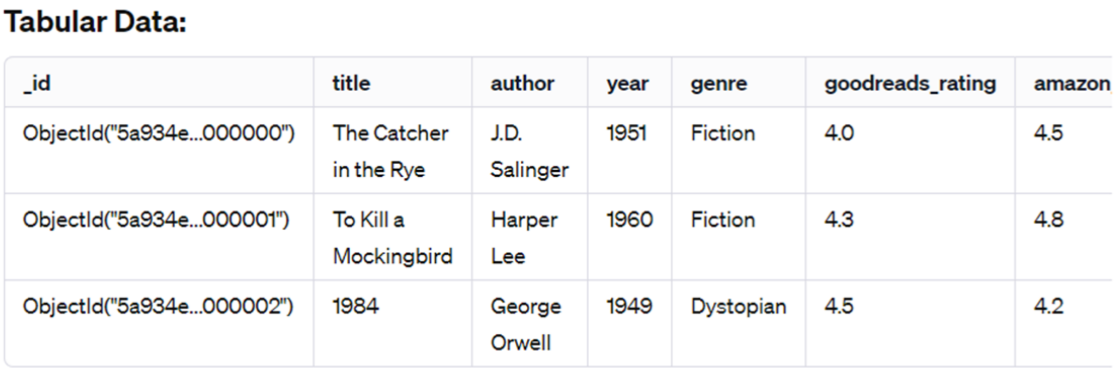
Ex: MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database, MariaDB, SQLite, IBM DB2

NoSQL databases, or non-relational databases, differ from traditional relational databases by not using tables for data storage. They are specifically designed for handling unstructured or semi-structured data, offering high performance and scalability. Some of the most common types of NoSQL databases include:



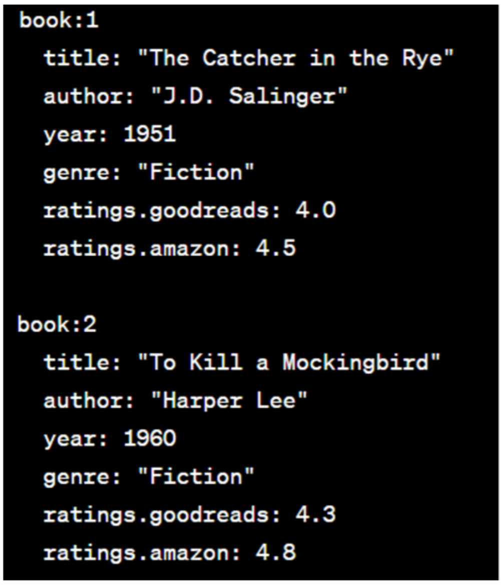
Key-value stores: Key-value stores store data as a collection of key-value pairs. This type of database is efficient for storing and retrieving data by key.

Ex: Redis, DynamoDB



Tabular Data:

_id	title	author	year	genre	goodreads_rating	amazon
ObjectId("5a934e...000000")	The Catcher in the Rye	J.D. Salinger	1951	Fiction	4.0	4.5
ObjectId("5a934e...000001")	To Kill a Mockingbird	Harper Lee	1960	Fiction	4.3	4.8
ObjectId("5a934e...000002")	1984	George Orwell	1949	Dystopian	4.5	4.2



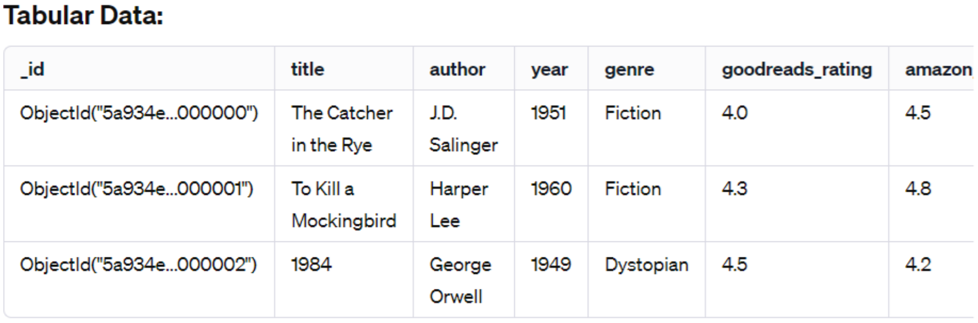
```
book:1
  title: "The Catcher in the Rye"
  author: "J.D. Salinger"
  year: 1951
  genre: "Fiction"
  ratings.goodreads: 4.0
  ratings.amazon: 4.5

book:2
  title: "To Kill a Mockingbird"
  author: "Harper Lee"
  year: 1960
  genre: "Fiction"
  ratings.goodreads: 4.3
  ratings.amazon: 4.8
```

Key-value stores:
Table Data is stored in the form of key value pair

In Redis, each book is represented as a hash, and each field in the hash corresponds to a column in the table. The book ID is used as part of the key to uniquely

Document stores: These databases store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages.

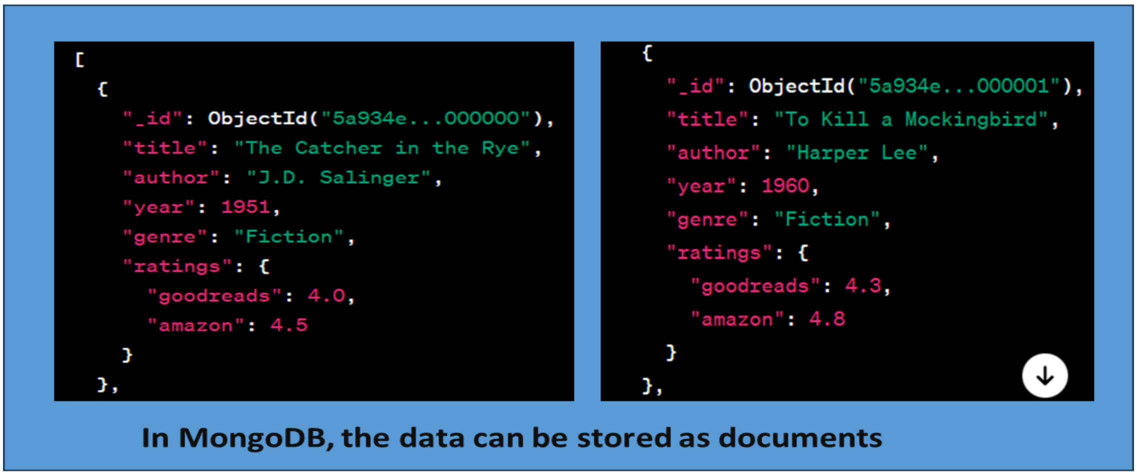
Ex: MongoDB, CouchDB



Tabular Data:

_id	title	author	year	genre	goodreads_rating	amazon
ObjectId("5a934e...000000")	The Catcher in the Rye	J.D. Salinger	1951	Fiction	4.0	4.5
ObjectId("5a934e...000001")	To Kill a Mockingbird	Harper Lee	1960	Fiction	4.3	4.8
ObjectId("5a934e...000002")	1984	George Orwell	1949	Dystopian	4.5	4.2

Document stores: Data stored in the form of JSON files



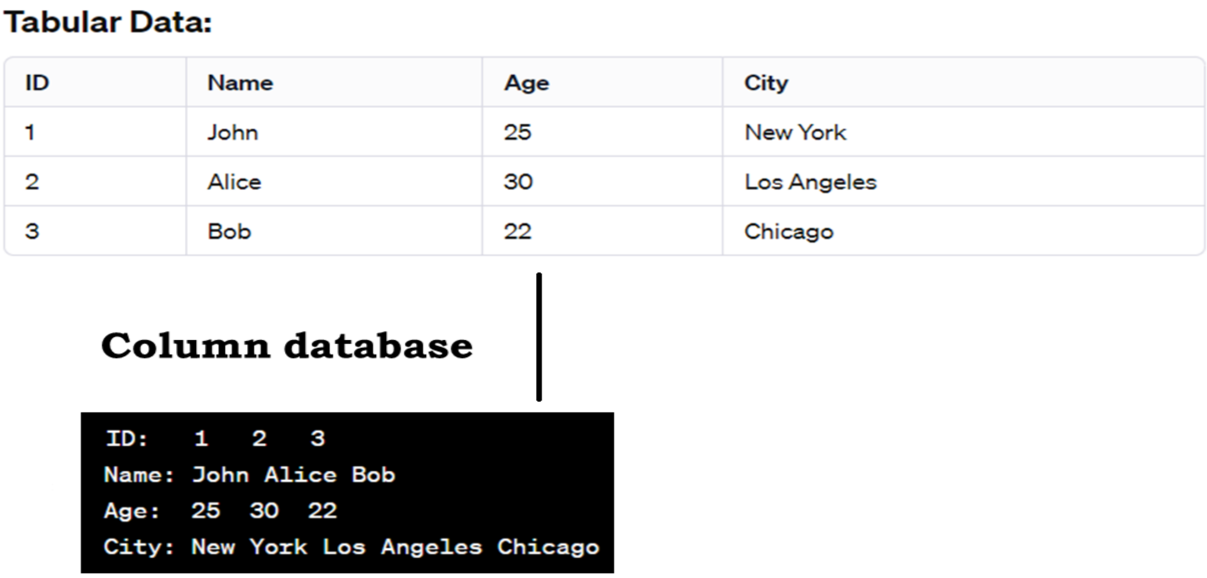
```
[
  {
    "_id": ObjectId("5a934e...000000"),
    "title": "The Catcher in the Rye",
    "author": "J.D. Salinger",
    "year": 1951,
    "genre": "Fiction",
    "ratings": {
      "goodreads": 4.0,
      "amazon": 4.5
    }
  },
```

```
  {
    "_id": ObjectId("5a934e...000001"),
    "title": "To Kill a Mockingbird",
    "author": "Harper Lee",
    "year": 1960,
    "genre": "Fiction",
    "ratings": {
      "goodreads": 4.3,
      "amazon": 4.8
    }
  },
```

In MongoDB, the data can be stored as documents

Column-family stores: These databases store data as column families, which are sets of columns that are treated as a single entity. They are optimized for fast and efficient querying of large amounts of data.

Ex: Apache Cassandra, HBase



Tabular Data:

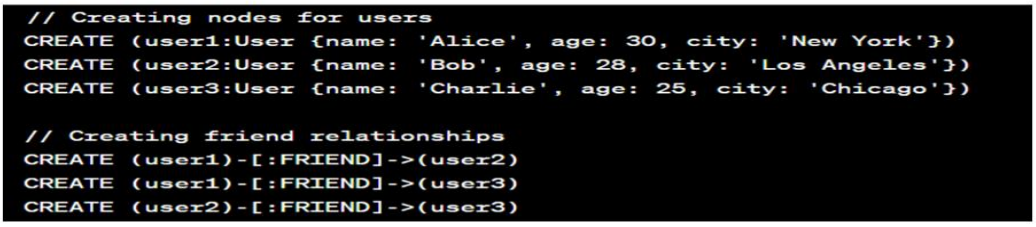
ID	Name	Age	City
1	John	25	New York
2	Alice	30	Los Angeles
3	Bob	22	Chicago

Column database

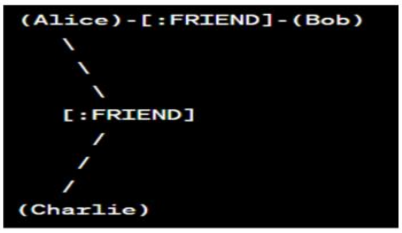
```
ID:    1   2   3
Name:  John Alice Bob
Age:   25  30  22
City:  New York Los Angeles Chicago
```

Graph databases: Graph databases store data as a collection of nodes and edges. Nodes represent entities, and edges represent relationships between entities.

Ex: Neo4j, Amazon Neptune



```
// Creating nodes for users
CREATE (user1:User {name: 'Alice', age: 30, city: 'New York'})
CREATE (user2:User {name: 'Bob', age: 28, city: 'Los Angeles'})
CREATE (user3:User {name: 'Charlie', age: 25, city: 'Chicago'})

// Creating friend relationships
CREATE (user1)-[:FRIEND]->(user2)
CREATE (user1)-[:FRIEND]->(user3)
CREATE (user2)-[:FRIEND]->(user3)
```



```
(Alice)-[:FRIEND]-(Bob)
    \
     \
      \
    [:FRIEND]
      /
     /
    /
(Charlie)
```

Graph database

Difference between SQL & NoSQL Databases

SQL	NoSQL
Databases are categorized as Relational Database Management System (RDBMS).	NoSQL databases are categorized as Non-relational or distributed database system.
SQL databases have fixed or static or predefined schema.	NoSQL databases have dynamic schema.
SQL databases display data in form of tables so it is known as table-based database.	NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores.
SQL databases are vertically scalable.	NoSQL databases are horizontally scalable.
SQL databases use a powerful language "Structured Query Language" to define and manipulate the data.	In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database.
SQL databases are best suited for complex queries.	NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries.
SQL databases are not best suited for hierarchical data storage.	NoSQL databases are best suited for hierarchical data storage.
Ex: MySQL, Oracle, Sqlite, PostgreSQL	Ex: MongoDB, Redis, Cassandra, Hbase, Neo4j, CouchDB etc.

Types of Variables:

Data consist of individuals and variables that give us information about those individuals. An individual can be an object or a person. A variable is an attribute, such as a measurement or a label.

Two types of Data

- Quantitative data
- Qualitative Data or Categorical data

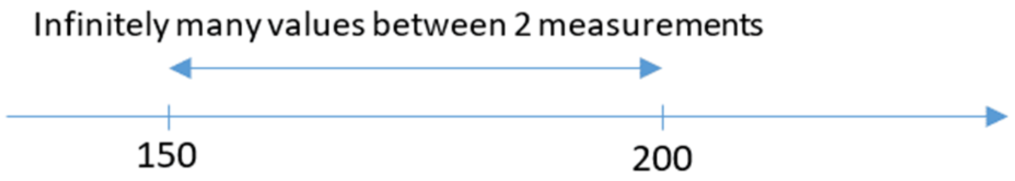
Qualitative Data: Qualitative data is non-numeric information that describes qualities or characteristics.

Quantitative Data: A quantitative or Numerical variable is a type of variable consisting of values that represent counts or measurements of a certain quantity. For instance, age, height, number of cigarettes smoked, etc.

A quantitative variable can be either continuous or discrete.

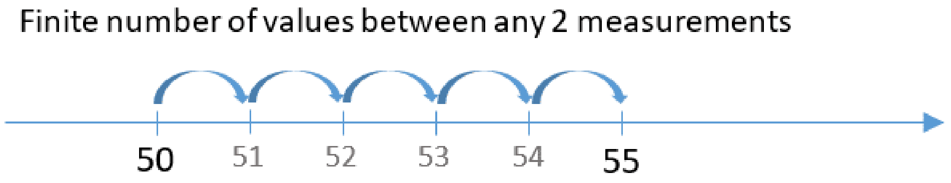
Continuous Variable:

- A continuous variable is a type of quantitative variable consisting of numerical values that can be measured but not counted, because there are infinitely many values between 1 measurement and another.
- EX: Weight, Distance, Volume, age etc.
- The weight of a person measured in kilograms (with decimal points) is a continuous ratio variable.



Discrete variable:

- A discrete variable is a type of quantitative variable consisting of numerical values that can be measured and counted, because these values are separate or distinct.
- Ex: Number of visits to the doctor , Number of Employees in a Company. Number of Students in a Classroom.



Is age discrete or continuous?

Age is a discrete variable when counted in years, for example when you ask someone about their age in a questionnaire. Age is a continuous variable when measured with high precision, for example when calculated from the exact date of birth.

Qualitative or Categorical variables: Categorical variables represent groupings of some kind. They are sometimes recorded as numbers, but the numbers represent categories rather than actual amounts of things. There are three types of categorical variables: **binary, nominal, and ordinal variables.**

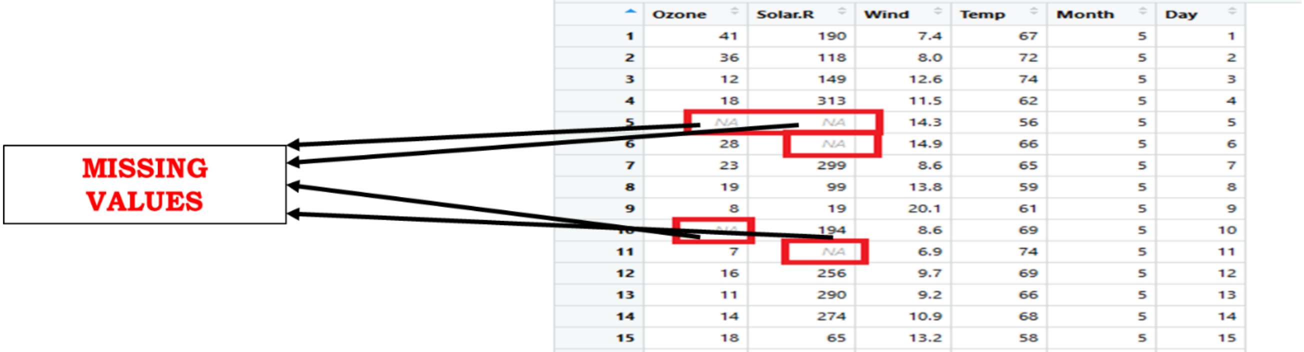
Type of variable	What does the data represent?	Examples
Binary variables	Yes/no outcomes.	•Heads/tails in a coin flip •Win/lose in a football game
Nominal variables	Groups with no rank or order between them.	•Colors •Brands •ZIP CODE
Ordinal variables	Groups that are ranked in a specific order.	•Finishing place in a race •Rating scale responses in a survey*

Handling Missing values in dataset using Imputation:

- Missing Values
- Deleting Missing values
- Handling Missing with mean/mode
- Handling Missing using MICE Package

Missing Values:

In statistics, missing data or missing values occurs when no data value is stored for the variable in an observation. Missing data are a common occurrence and can have a significant effect on the conclusions that can be drawn from the data.



	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
5	NA	NA	14.3	56	5	5
6	28	NA	14.9	66	5	6
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9
10	NA	194	8.6	69	5	10
11	7	NA	6.9	74	5	11
12	16	256	9.7	69	5	12
13	11	290	9.2	66	5	13
14	14	274	10.9	68	5	14
15	18	65	13.2	58	5	15

The many methods, proposed over the years, to handle missing values can be separated in two main groups:

- Deletion and
- Imputation.

There are three common deletion approaches:

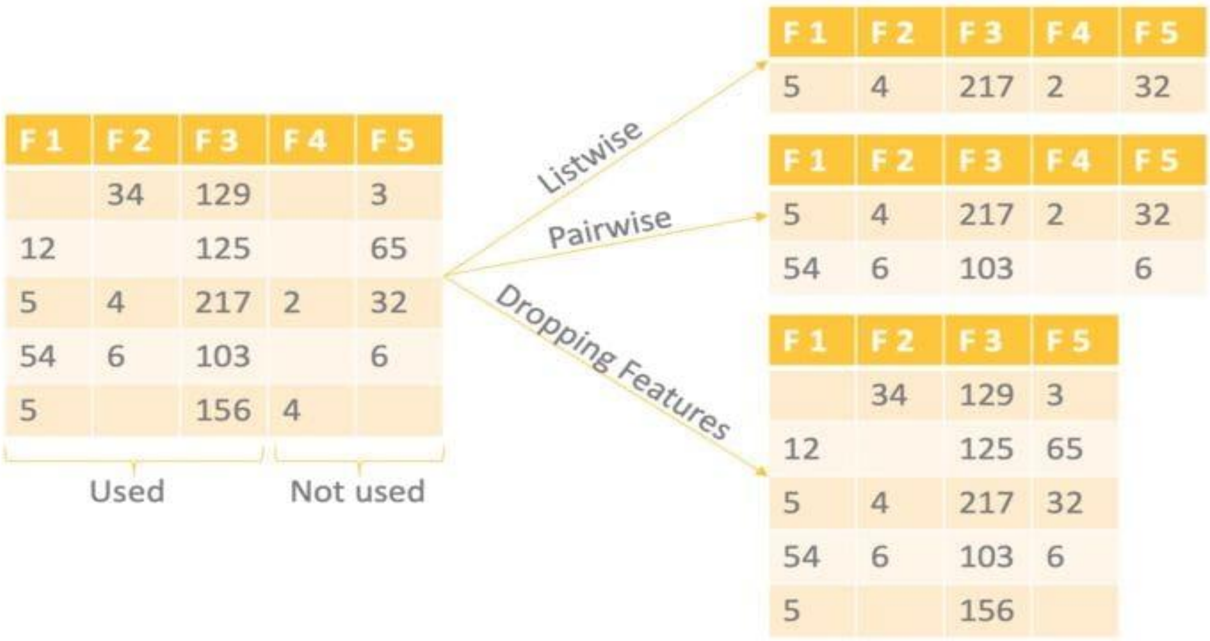
- Listwise deletion
- Pairwise deletion, and

- Dropping features.

Listwise Deletion: Delete all rows where one or more values are missing.

Pairwise Deletion: Delete only the rows that have missing values in the columns used for the analysis.

Dropping Features: Drop entire columns with more missing values.



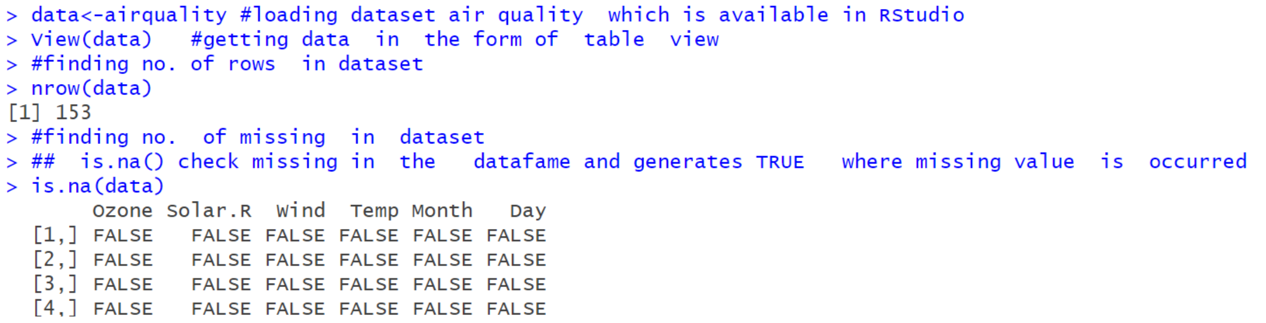
F1	F2	F3	F4	F5
	34	129		3
12		125		65
5	4	217	2	32
54	6	103		6
5		156	4	

F1	F2	F3	F4	F5
5	4	217	2	32

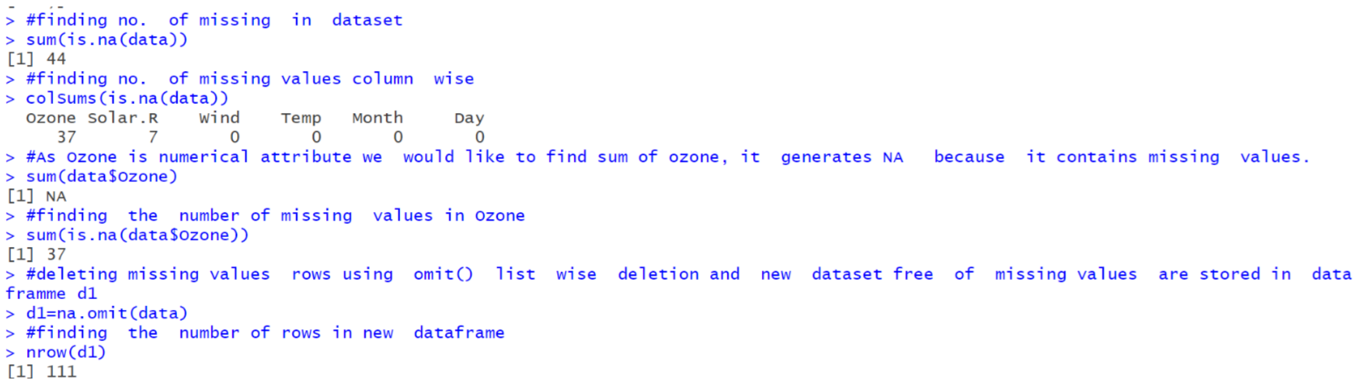
F1	F2	F3	F4	F5
5	4	217	2	32
54	6	103		6

F1	F2	F3	F5
	34	129	3
12		125	65
5	4	217	32
54	6	103	6
5		156	

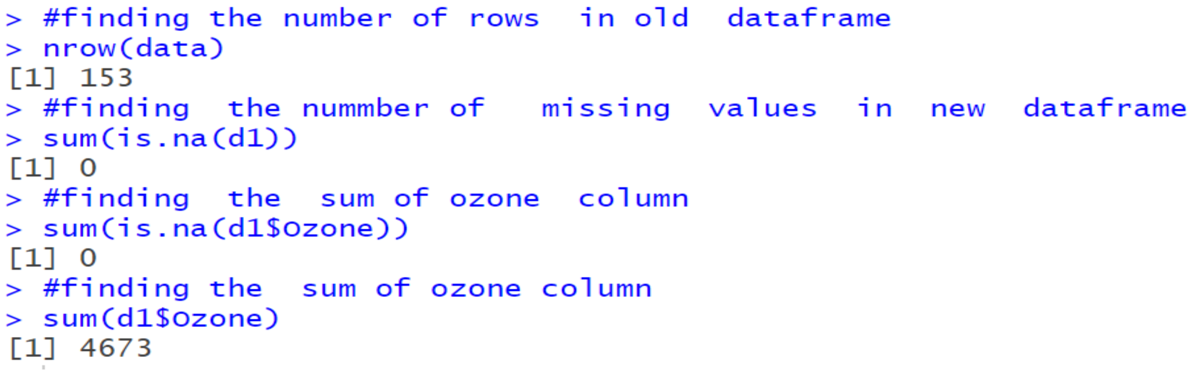
R Code for Deleting missing values in airquality dataset



```
> data<-airquality #loading dataset air quality  which is available in RStudio
> View(data)   #getting data  in  the form of  table  view
> #finding no. of rows  in dataset
> nrow(data)
[1] 153
> #finding no.  of missing  in  dataset
> ##  is.na() check missing in  the   datafame and generates TRUE   where missing value  is  occurred
> is.na(data)
        Ozone Solar.R  Wind  Temp Month   Day
  [1,] FALSE    FALSE FALSE FALSE FALSE FALSE
  [2,] FALSE    FALSE FALSE FALSE FALSE FALSE
  [3,] FALSE    FALSE FALSE FALSE FALSE FALSE
  [4,] FALSE    FALSE FALSE FALSE FALSE FALSE
```



```
> #finding no.  of missing  in  dataset
> sum(is.na(data))
[1] 44
> #finding no.  of missing values column  wise
> colSums(is.na(data))
  Ozone Solar.R    Wind    Temp   Month     Day 
     37       7       0       0       0       0 
> #As Ozone is numerical attribute we  would like to find sum of ozone, it  generates NA   because  it contains missing  values.
> sum(data$Ozone)
[1] NA
> #finding  the  number of missing  values in Ozone
> sum(is.na(data$Ozone))
[1] 37
> #deleting missing values  rows using  omit()  list  wise  deletion and  new  dataset free  of  missing values  are stored in  data framme d1
> d1=na.omit(data)
> #finding  the  number of rows in new  dataframe
> nrow(d1)
[1] 111
```

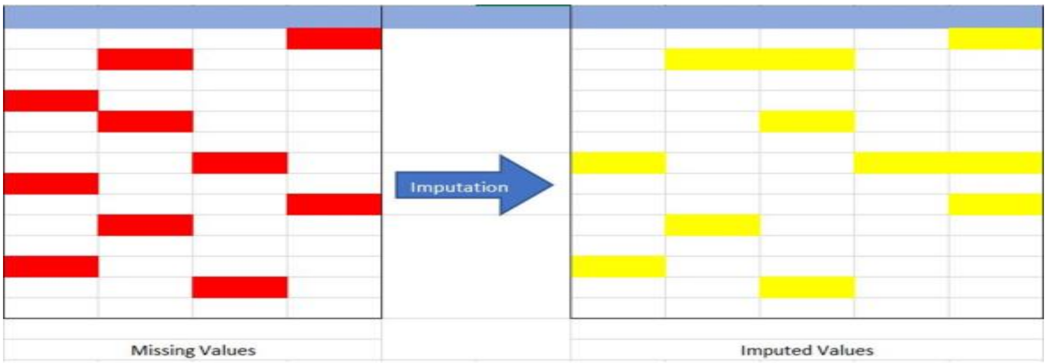


```
> #finding the number of rows  in old  dataframe
> nrow(data)
[1] 153
> #finding  the nummber of   missing  values  in  new  dataframe
> sum(is.na(d1))
[1] 0
> #finding  the  sum of ozone  column
> sum(is.na(d1$Ozone))
[1] 0
> #finding the  sum of ozone column
> sum(d1$Ozone)
[1] 4673
```

Missing Imputation:

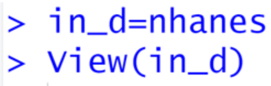
Imputation is a technique used for replacing the missing data with some substitute value to retain most of the data/information of the dataset.

These techniques are used because removing the data from the dataset every time is not feasible and can lead to a reduction in the size of the dataset to a large extend, and also leads to incorrect analysis.



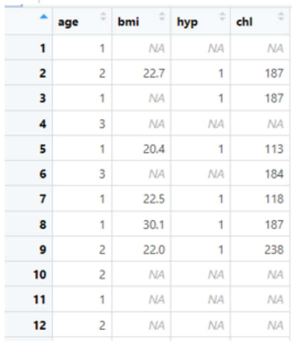
Imputed value, is **an assumed value given to an item when the actual value is not known or available**

Handling missing values: Use attribute mean to fill in the missing value

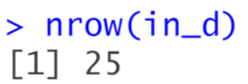


```
> in_d=nhanes
> View(in_d)
```

Loading a predefined dataset & viewing a predefined dataset

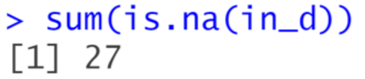


	age	bmi	hyp	chl
1	1	NA	NA	NA
2	2	22.7	1	187
3	1	NA	1	187
4	3	NA	NA	NA
5	1	20.4	1	113
6	3	NA	NA	184
7	1	22.5	1	118
8	1	30.1	1	187
9	2	22.0	1	238
10	2	NA	NA	NA
11	1	NA	NA	NA
12	2	NA	NA	NA



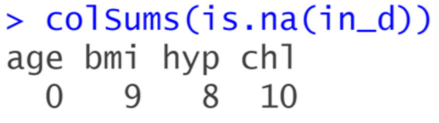
```
> nrow(in_d)
[1] 25
```

# finding number of rows



```
> sum(is.na(in_d))
[1] 27
```

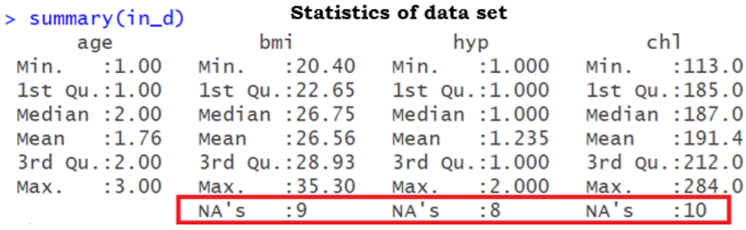
# finding number of missing values in complete dataset



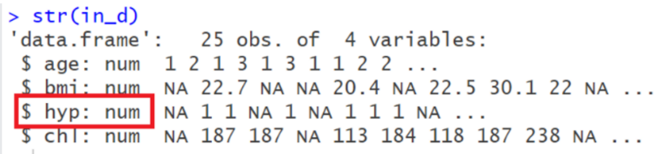
```
> colSums(is.na(in_d))
age bmi hyp chl 
  0   9   8  10 
```

# finding missing values in each and every column of dataset

Statistics of data set



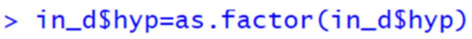
```
> summary(in_d)
      age            bmi             hyp             chl       
 Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0  
 1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0  
 Median :2.00   Median :26.75   Median :1.000   Median :187.0  
 Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4  
 3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0  
 Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0  
                NA's   :9       NA's   :8       NA's   :10     
```



```
> str(in_d)
'data.frame':   25 obs. of  4 variables:
 $ age: num  1 2 1 3 1 3 1 1 2 2 ...
 $ bmi: num  NA 22.7 NA NA 20.4 NA 22.5 30.1 22 NA ...
 $ hyp: num  NA 1 1 NA 1 NA 1 1 1 NA ...
 $ chl: num  NA 187 187 NA 113 184 118 187 238 NA ...
```

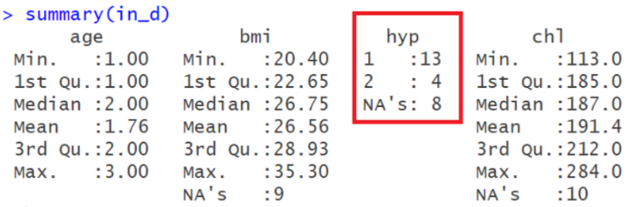
# finding structure of dataset

Age, bmi & chl are numerical attributes(DISCRETE) where as hyp is not numeric(DISCRETE), We can convert it to categorical

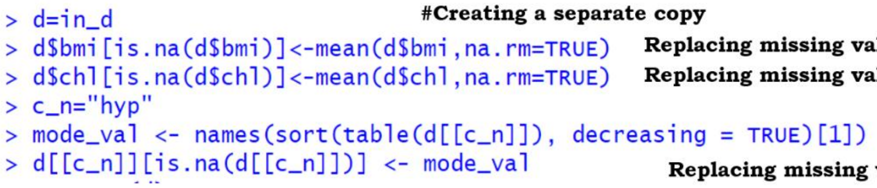


```
> in_d$hyp=as.factor(in_d$hyp)
```

#Converting into categorical



```
> summary(in_d)
      age             bmi           hyp          chl       
 Min.   :1.00   Min.   :20.40   1   :13   Min.   :113.0  
 1st Qu.:1.00   1st Qu.:22.65   2   : 4   1st Qu.:185.0  
 Median :2.00   Median :26.75   NA's: 8   Median :187.0  
 Mean   :1.76   Mean   :26.56             Mean   :191.4  
 3rd Qu.:2.00   3rd Qu.:28.93             3rd Qu.:212.0  
 Max.   :3.00   Max.   :35.30             Max.   :284.0  
                NA's   :9                 NA's   :10     
```



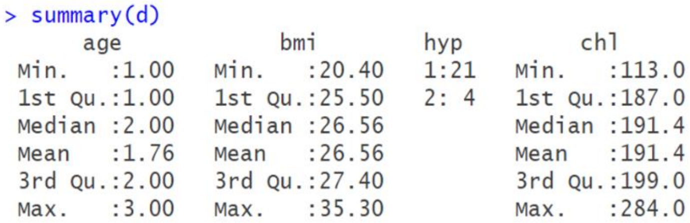
```
> d=in_d
> d$bmi[is.na(d$bmi)]<-mean(d$bmi,na.rm=TRUE)
> d$chl[is.na(d$chl)]<-mean(d$chl,na.rm=TRUE)
> c_n="hyp"
> mode_val <- names(sort(table(d[[c_n]]), decreasing = TRUE)[1])
> d[[c_n]][is.na(d[[c_n]])] <- mode_val
```

#Creating a separate copy

Replacing missing values with mean of bmi attribute

Replacing missing values with mean of chl attribute

Replacing missing values with mode of hyp attribute



```
 > summary(d)
      age             bmi          hyp         chl       
 Min.   :1.00   Min.   :20.40   1:21   Min.   :113.0  
 1st Qu.:1.00   1st Qu.:25.50   2: 4   1st Qu.:187.0  
 Median :2.00   Median :26.56          Median :191.4  
 Mean   :1.76   Mean   :26.56          Mean   :191.4  
 3rd Qu.:2.00   3rd Qu.:27.40          3rd Qu.:199.0  
 Max.   :3.00   Max.   :35.30          Max.   :284.0  
```

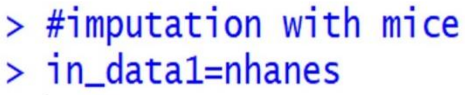
Statistics after handling missing values

MICE Package : Multiple Imputation by Chained Equations

- MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package in R.
- MICE assumes that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on observed value and can be predicted using them.
- For example: Suppose we have X1, X2….Xk variables. If X1 has missing values, then it will be regressed on other variables X2 to Xk. The missing values in X1 will be then replaced by predictive values obtained. Similarly, if X2 has missing values, then X1, X3 to Xk variables will be used in prediction model as independent variables
- By default, linear regression is used to predict continuous missing values. Logistic regression is used for categorical missing values. Once this cycle is complete, multiple data sets are generated. These data sets differ only in imputed missing values.

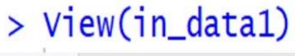
Precisely, the methods used by this package are:

- PMM (Predictive Mean Matching) – For numeric variables
- logreg(Logistic Regression) – For Binary Variables(with 2 levels)
- polyreg(Bayesian polytomous regression) – For Factor Variables (>= 2 levels)
- Proportional odds model (ordered, >= 2 levels)



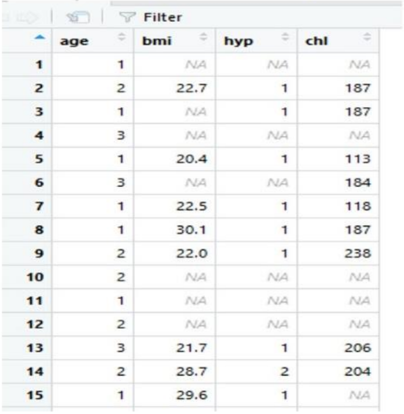
```
> #imputation with mice
> in_data1=nhanes
```

#Loading a predefined data set that comes with MICE Package

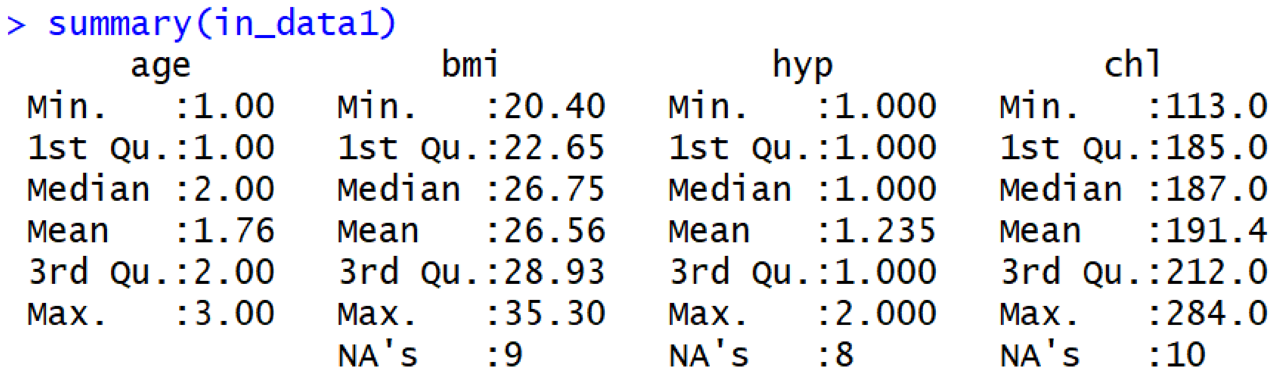


```
> View(in_data1)
```

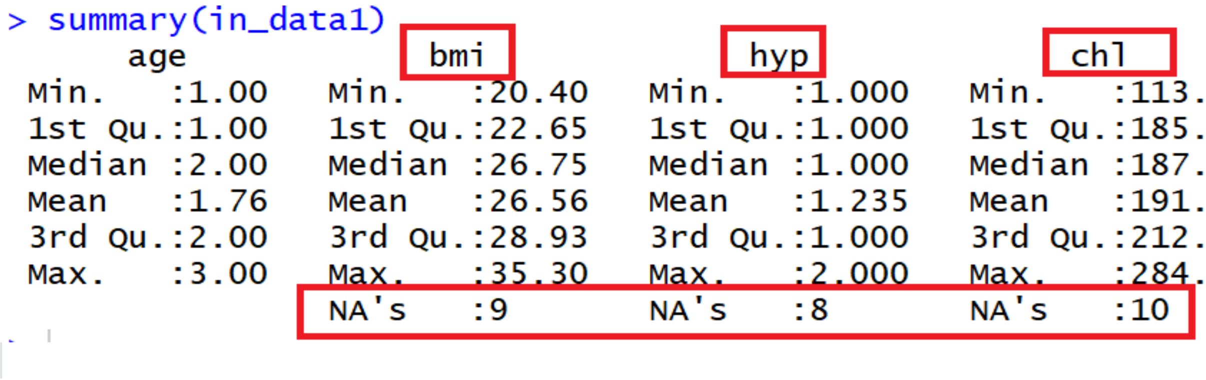
#Viewing data set, it contains three four columns and 25 rows



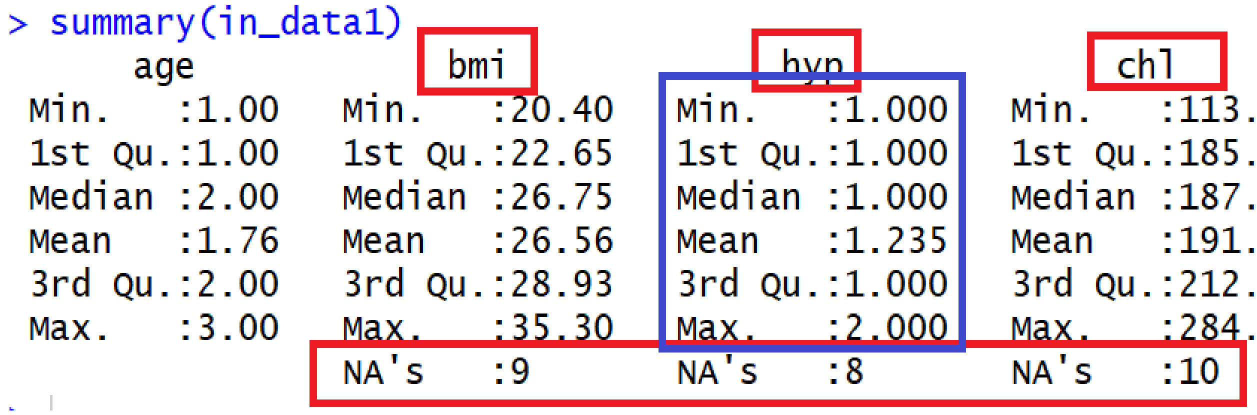
	age	bmi	hyp	chl
1	1	NA	NA	NA
2	2	22.7	1	187
3	1	NA	1	187
4	3	NA	NA	NA
5	1	20.4	1	113
6	3	NA	NA	184
7	1	22.5	1	118
8	1	30.1	1	187
9	2	22.0	1	238
10	2	NA	NA	NA
11	1	NA	NA	NA
12	2	NA	NA	NA
13	3	21.7	1	206
14	2	28.7	2	204
15	1	29.6	1	NA



```
> summary(in_data1)
      age             bmi             hyp             chl
 Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0
 1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0
 Median :2.00   Median :26.75   Median :1.000   Median :187.0
 Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4
 3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0
 Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0
                NA's   :9       NA's   :8       NA's   :10
```



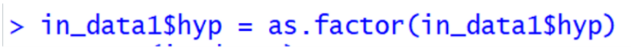
```
> summary(in_data1)
      age             bmi             hyp             chl
 Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0
 1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0
 Median :2.00   Median :26.75   Median :1.000   Median :187.0
 Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4
 3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0
 Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0
                NA's   :9       NA's   :8       NA's   :10
```



```
> summary(in_data1)
      age            bmi              hyp              chl
 Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0
 1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0
 Median :2.00   Median :26.75   Median :1.000   Median :187.0
 Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4
 3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0
 Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0
                NA's   :9       NA's   :8       NA's   :10
```

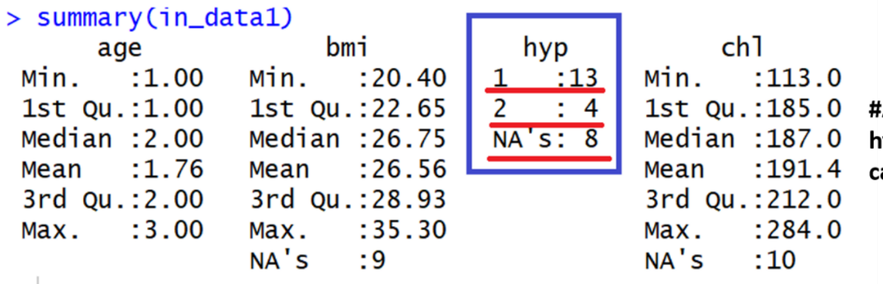
Hyp is categorical variable, for categorical variable calculated interquartile range and mean and median is of no use.

Instead of that counting no of entries belongs to category 1 and category 2 is better.



```
> in_data1$hyp = as.factor(in_data1$hyp)
```

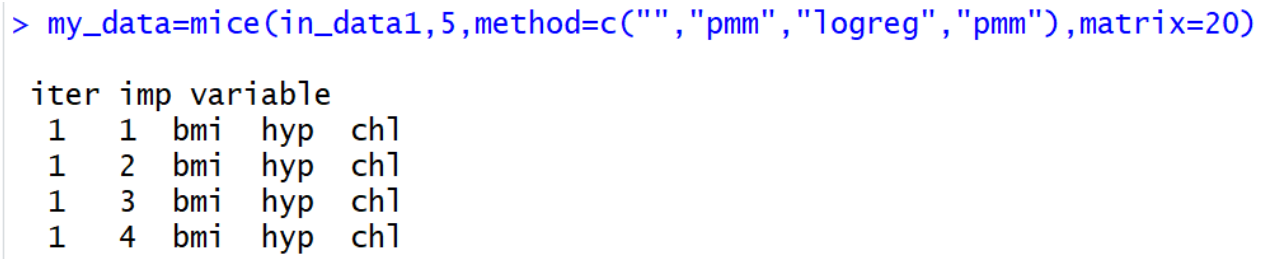
#Converts the hyp variable to Categorical Variable



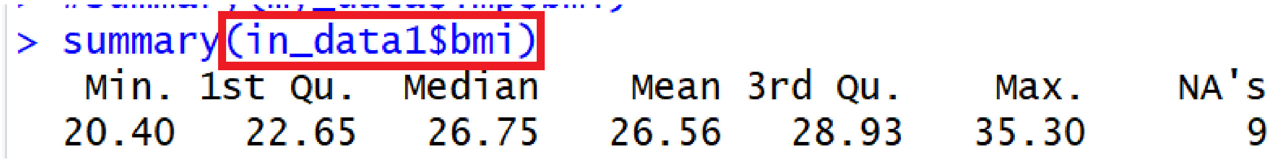
```
> summary(in_data1)
      age            bmi           hyp          chl
 Min.   :1.00   Min.   :20.40   1   :13   Min.   :113.0
 1st Qu.:1.00   1st Qu.:22.65   2   : 4   1st Qu.:185.0
 Median :2.00   Median :26.75   NA's: 8   Median :187.0
 Mean   :1.76   Mean   :26.56             Mean   :191.4
 3rd Qu.:2.00   3rd Qu.:28.93             3rd Qu.:212.0
 Max.   :3.00   Max.   :35.30             Max.   :284.0
                NA's   :9                 NA's   :10
```

#After applying as.factor() function on hyp variable, it was converted into categorical variable.

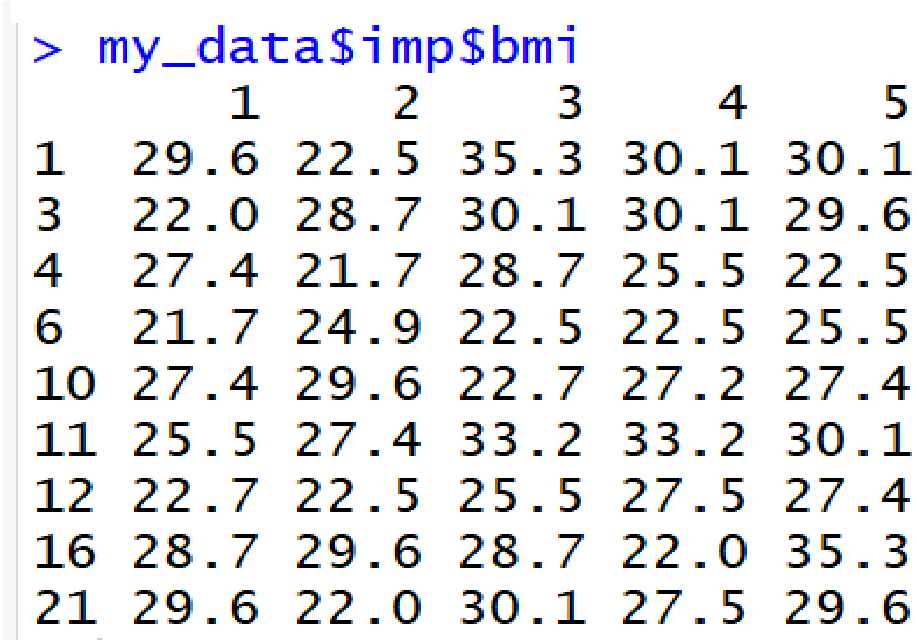
Mice() function missing values and generates five dataset among five data sets we can use any one based on mean comparing



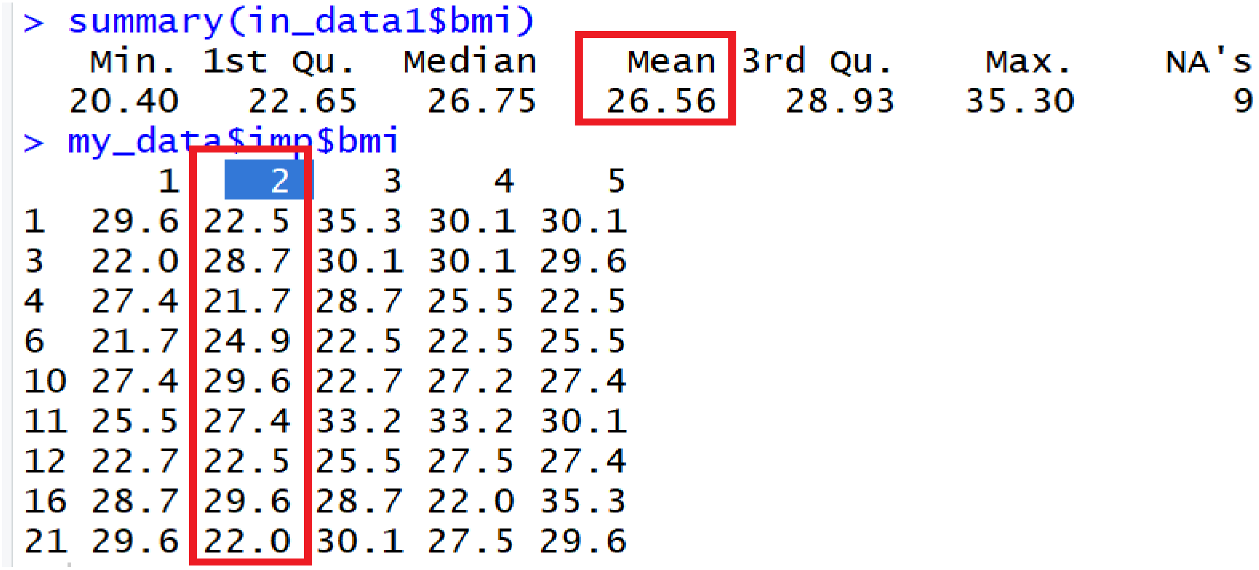
```
> my_data=mice(in_data1,5,method=c("","pmm","logreg","pmm"),matrix=20)

 iter imp variable
  1   1  bmi  hyp  chl
  1   2  bmi  hyp  chl
  1   3  bmi  hyp  chl
  1   4  bmi  hyp  chl
```



```
> summary(in_data1$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  20.40   22.65   26.75   26.56   28.93   35.30       9 
```

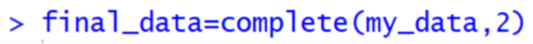


```
> my_data$imp$bmi
      1    2    3    4    5
1  29.6 22.5 35.3 30.1 30.1
3  22.0 28.7 30.1 30.1 29.6
4  27.4 21.7 28.7 25.5 22.5
6  21.7 24.9 22.5 22.5 25.5
10 27.4 29.6 22.7 27.2 27.4
11 25.5 27.4 33.2 33.2 30.1
12 22.7 22.5 25.5 27.5 27.4
16 28.7 29.6 28.7 22.0 35.3
21 29.6 22.0 30.1 27.5 29.6
```



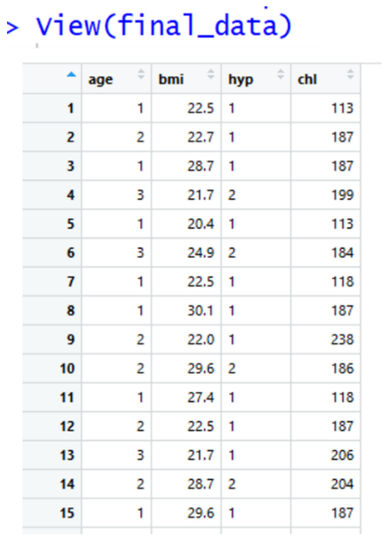
```
> summary(in_data1$bmi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  20.40   22.65   26.75   26.56   28.93   35.30       9 
> my_data$imp$bmi
      1    2    3    4    5
1  29.6 22.5 35.3 30.1 30.1
3  22.0 28.7 30.1 30.1 29.6
4  27.4 21.7 28.7 25.5 22.5
6  21.7 24.9 22.5 22.5 25.5
10 27.4 29.6 22.7 27.2 27.4
11 25.5 27.4 33.2 33.2 30.1
12 22.7 22.5 25.5 27.5 27.4
16 28.7 29.6 28.7 22.0 35.3
21 29.6 22.0 30.1 27.5 29.6
```

By comparing mean of BMI of old data set and new data set, we can conclude that 2nd data set in my_data variable is effective and we can consider that dataset.



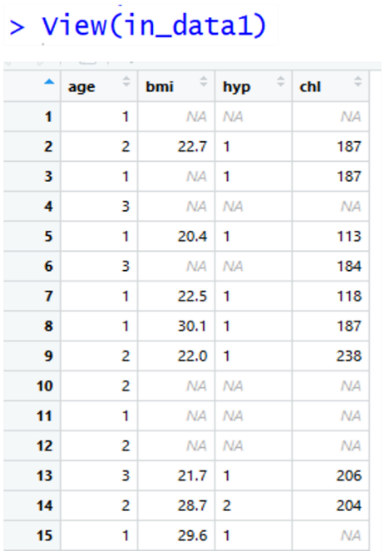
```
> final_data=complete(my_data,2)
```

Generating final data from mice



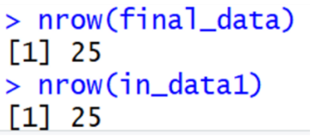
```
> View(final_data)
```

	age	bmi	hyp	chl
1	1	22.5	1	113
2	2	22.7	1	187
3	1	28.7	1	187
4	3	21.7	2	199
5	1	20.4	1	113
6	3	24.9	2	184
7	1	22.5	1	118
8	1	30.1	1	187
9	2	22.0	1	238
10	2	29.6	2	186
11	1	27.4	1	118
12	2	22.5	1	187
13	3	21.7	1	206
14	2	28.7	2	204
15	1	29.6	1	187



```
> View(in_data1)
```

	age	bmi	hyp	chl
1	1	NA	NA	NA
2	2	22.7	1	187
3	1	NA	1	187
4	3	NA	NA	NA
5	1	20.4	1	113
6	3	NA	NA	184
7	1	22.5	1	118
8	1	30.1	1	187
9	2	22.0	1	238
10	2	NA	NA	NA
11	1	NA	NA	NA
12	2	NA	NA	NA
13	3	21.7	1	206
14	2	28.7	2	204
15	1	29.6	1	NA



```
> nrow(final_data)
[1] 25
> nrow(in_data1)
[1] 25
```

Data Modeling Techniques:

In Artificial Intelligence, a model is an abstract representation of a decision process. Its primary goal is to enable the decision process automation, often applied to business. The model can also help understanding the modeled process itself. Machine Learning models are mathematical algorithms that are "trained" using data. Ideally, the model should also explain the reason behind its decision to help understand the decision process.

Predictive modeling: In predictive modeling, the outcome defines the performed task. If the outcome is made of continuous values, it is a regression task. The model will then return a numerical value.

If the outcome is made of two or more categories, it is a classification task. The model will then deliver a class. When there are two classes, we talk about binary classification and multi-classification otherwise.

Several algorithms can perform regression and classification tasks to build predictive models: regression algorithms, Bayesian algorithms, kernel algorithms, decision trees, neural networks, and evolutionary algorithms such as ZGP (the core engine of MyDataModels products).

- ➢ Regression analysis mainly focuses on finding a relationship between a dependent variable and one or more independent variables.

- Predict the value of a dependent variable based on the value of at least one independent variable.
- It explains the impact of changes in an independent variable on the dependent variable.

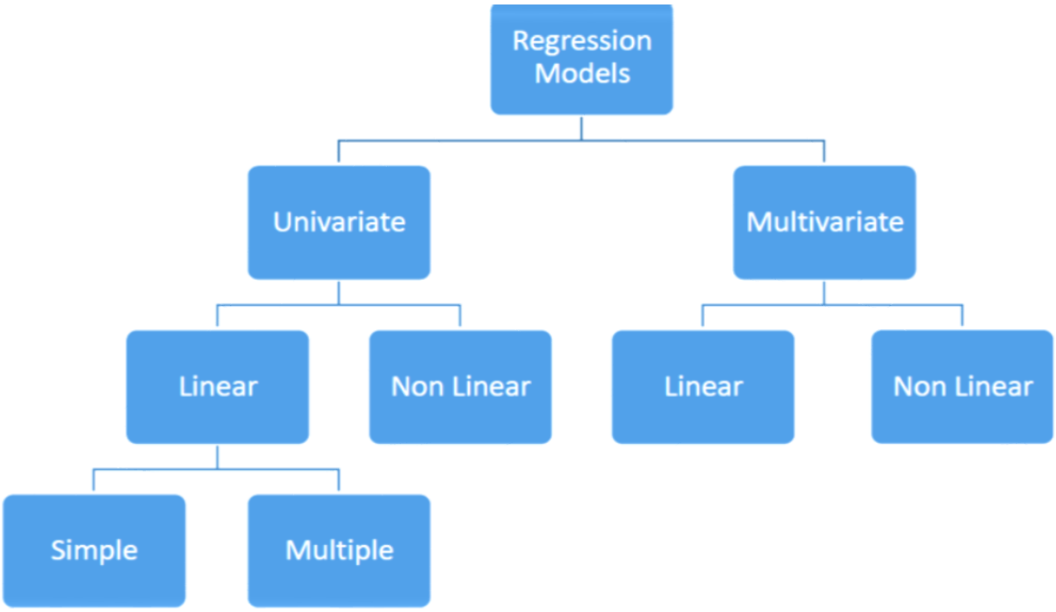
  $$Y = f(X, \beta)$$

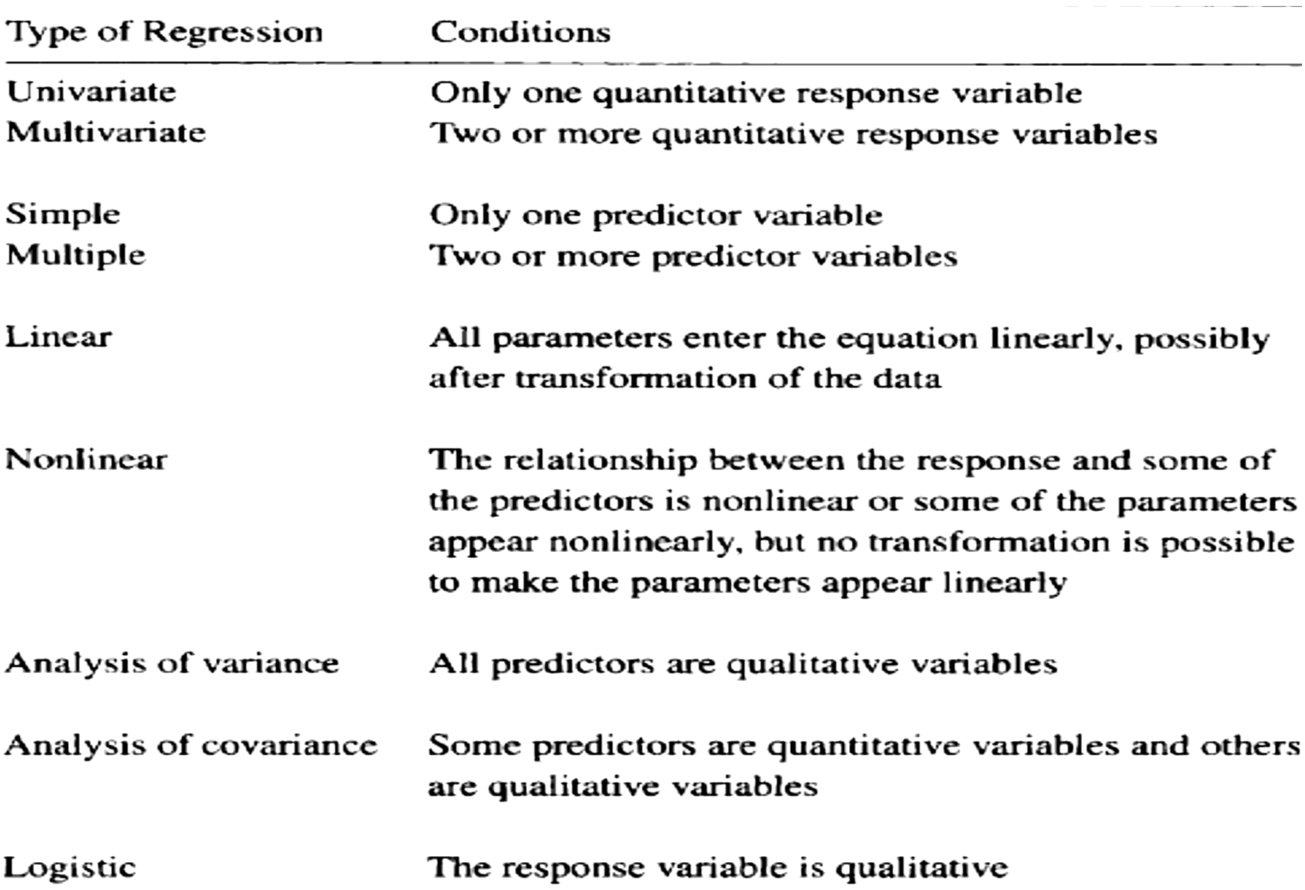
  where Y is the dependent variable

  X is the independent variable

  β is the unknown coefficient

- Widely used in prediction and forecasting





Type of Regression	Conditions
Univariate	Only one quantitative response variable
Multivariate	Two or more quantitative response variables
Simple	Only one predictor variable
Multiple	Two or more predictor variables
Linear	All parameters enter the equation linearly, possibly after transformation of the data
Nonlinear	The relationship between the response and some of the predictors is nonlinear or some of the parameters appear nonlinearly, but no transformation is possible to make the parameters appear linearly
Analysis of variance	All predictors are qualitative variables
Analysis of covariance	Some predictors are quantitative variables and others are qualitative variables
Logistic	The response variable is qualitative

Need for Business Modelling:

The main need of Business Modelling for the Companies that embrace big data analytics and transform their business models in parallel will create new opportunities for revenue streams, customers, products and services.

Application of Modelling in Business:

- Applications of Data Modelling can be termed as Business analytics.
- Business analytics involves the collating, sorting, processing, and studying of business-related data using statistical models and iterative methodologies. The goal of BA is to narrow down which datasets are useful and which can increase revenue, productivity, and efficiency.
- Business analytics (BA) is the combination of skills, technologies, and practices used to examine an organization's data and performance as a way to gain insights and make data-driven decisions in the future using statistical analysis. Although business analytics is being leveraged in most commercial sectors and industries, the following applications are the most common.

1. Credit Card Companies: Credit and debit cards are an everyday part of consumer spending, and they are an ideal way of gathering information about a purchaser's spending habits, financial situation, behavior trends, demographics, and lifestyle preferences.

2. Customer Relationship Management (CRM) Excellent customer relations is critical for any company that wants to retain customer loyalty to stay in business for the long haul. CRM systems analyze important performance indicators such as demographics, buying patterns, socio-economic information, and lifestyle.

3. Finance The financial world is a volatile place, and business analytics helps to extract insights that help organizations maneuver their way through tricky terrain. Corporations turn to business analysts to optimize budgeting, banking, financial planning, forecasting, and portfolio management.

4. Human Resources Business analysts help the process by pouring through data that characterizes high performing candidates, such as educational background, attrition rate, the average length of employment, etc. By working with this information, business analysts help HR by forecasting the best fits between the company and candidates.

5. Manufacturing Business analysts work with data to help stakeholders understand the things that affect operations and the bottom line. Identifying things like equipment downtime, inventory levels, and maintenance costs help companies streamline inventory management, risks, and supply-chain management to create maximum efficiency.

6. Marketing Business analysts help answer these questions and so many more, by measuring marketing and advertising metrics, identifying consumer behavior and the target audience, and analyzing market trends.