**Statistical Inference**

Statistical inference is the process of analysing the result and making conclusions from data subject to random variation. It is also called inferential statistics. Hypothesis testing and [confidence intervals](https://byjus.com/maths/confidence-interval/)are the applications of the statistical inference. Statistical inference is a method of making decisions about the parameters of a population, based on random sampling. It helps to assess the relationship between the dependent and independent variables. The purpose of statistical inference to estimate the uncertainty or sample to sample variation. It allows us to provide a probable range of values for the true values of something in the population. The components used for making statistical inference are:

* Sample Size
* Variability in the sample
* Size of the observed differences

**Types of Statistical Inference**

There are different types of statistical inferences that are extensively used for making conclusions. They are:

* One sample hypothesis testing
* Confidence Interval
* Pearson Correlation
* Bi-variate regression
* Multi-variate regression
* Chi-square statistics and contingency table
* ANOVA or T-test

**Statistical Inference Procedure**

The procedure involved in inferential statistics are:

* Begin with a theory
* Create a research hypothesis
* Operationalize the variables
* Recognize the population to which the study results should apply
* Formulate a null hypothesis for this population
* Accumulate a sample from the population and continue the study
* Conduct statistical tests to see if the collected sample properties are adequately different from what would be expected under the [null hypothesis](https://byjus.com/maths/null-hypothesis/) to be able to reject the null hypothesis

**Statistical Inference Solution**

Statistical inference solutions produce efficient use of statistical data relating to groups of individuals or trials. It deals with all characters, including the collection, investigation and analysis of data and organizing the collected data. By statistical inference solution, people can acquire knowledge after starting their work in diverse fields. Some statistical inference solution facts are:

* It is a common way to assume that the observed sample is of independent observations from a population type like Poisson or normal
* Statistical inference solution is used to evaluate the parameter(s) of the expected model like normal mean or binomial proportion

**Importance of Statistical Inference**

Inferential Statistics is important to examine the data properly. To make an accurate conclusion, proper data analysis is important to interpret the research results. It is majorly used in the future prediction for various observations in different fields. It helps us to make inference about the data. The statistical inference has a wide range of application in different fields, such as:

* Business Analysis
* Artificial Intelligence
* Financial Analysis
* Fraud Detection
* Machine Learning
* Share Market
* Pharmaceutical Sector

R Data types are used in computer programming to specify the kind of data that can be stored in a variable. For effective memory consumption and precise computation, the right data type must be selected. Each R data type has its own set of regulations and restrictions.

**Data Types in R Programming Language**

Each variable in R has an associated data type. Each R-Data Type requires different amounts of memory and has some specific operations which can be performed over it. [R Programming language](https://www.geeksforgeeks.org/r-programming-language-introduction/) has the following basic R-data types and the following table shows the data type and the values that each data type can take.

| **Basic Data Types** | **Values** | **Examples** |
| --- | --- | --- |
| Numeric | Set of all real numbers | "numeric\_value <- 3.14" |
| Integer | Set of all integers, Z | "integer\_value <- 42L" |
| Logical | TRUE and FALSE | "logical\_value <- TRUE" |
| Complex | Set of complex numbers | "complex\_value <- 1 + 2i" |
| Character | “a”, “b”, “c”, …, “@”, “#”, “$”, …., “1”, “2”, …etc | "character\_value <- "Hello Geeks" |
| raw | as.raw() | "single\_raw <- as.raw(255)" |

**Numeric Data type in R**

Decimal values are called numerics in R. It is the default  R data type for numbers in R. If you assign a decimal value to a variable x as follows, x will be of numeric type. Real numbers with a decimal point are represented using this data type in R. it uses a format for double-precision floating-point numbers to represent numerical values.

### Integer Data type in R

R supports integer data types which are the set of all integers. You can create as well as convert a value into an integer type using the **as.integer()** function. You can also use the capital ‘L’ notation as a suffix to denote that a particular value is of the integer R data type.

### Logical Data type in R

R has logical data types that take either a value of true or false. A logical value is often created via a comparison between variables. Boolean values, which have two possible values, are represented by this R data type: FALSE or TRUE

### Complex Data type in R

R supports complex data types that are set of all the complex numbers. The complex data type is to store numbers with an imaginary component.

### Character Data type in R

R supports character data types where you have all the alphabets and special characters. It stores character values or strings. Strings in R can contain alphabets, numbers, and symbols. The easiest way to denote that a value is of character type in R data type is to wrap the value inside single or double inverted commas.

### ****Raw data type in R****

To save and work with data at the byte level in R, use the raw data type. By displaying a series of unprocessed bytes, it enables low-level operations on binary data. Here are some speculative data on R’s raw data types:

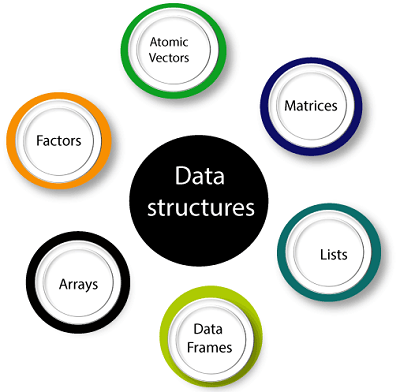
### Find data type of an object in R

To find the data type of an object you have to use **class()** function. The syntax for doing that is you need to pass the object as an argument to the function **class()** to find the data type of an object.

**Data Structures in R Programming**

Data structures are very important to understand. Data structure are the objects which we will manipulate in our day-to-day basis in R. Dealing with object conversions is the most common sources of despairs for beginners. We can say that everything in R is an object.

R has many data structures, which include:



1. Atomic vector
2. List
3. Array
4. Matrices
5. Data Frame
6. Factors

Vectors

A vector is the basic data structure in R, or we can say vectors are the most basic R data objects. There are six types of atomic vectors such as logical, integer, character, double, and raw. **"A vector is a collection of elements which is most commonly of mode character, integer, logical or numeric"** A vector can be one of the following two types:

1. Atomic vector
2. Lists

List

In R, **the list** is the container. Unlike an atomic vector, the list is not restricted to be a single mode. A list contains a mixture of data types. The list is also known as generic vectors because the element of the list can be of any type of R object. **"A list is a special type of vector in which each element can be a different type."**

We can create a list with the help of list() or as.list(). We can use vector() to create a required length empty list.

A list is generated using **list()** function. It is basically a generic vector that contains different objects.

R allows its users to perform **various operations** on lists which can be used to illustrate the data in different forms.

#### Creating a List

Lists in R can be created by placing the sequence inside the**list()**function.

#### Naming the elements of a list

Name can be assigned to the elements of the list and those names can be used to access the elements.

#### Accessing elements of a List

In order to access the list elements, use the index number, and in case of named list, elements can be accessed using its name also.

#### Merging elements of a List

Many lists can be merged in one list by which all the list elements are placed inside one list.

#### ****Converting a list to vector****

In order to perform arithmetic operations, lists should be converted to vectors using**unlist()** function.

#### ****Adding, Deleting****,****and Updating elements of a list****

In R, a new element can be added to the list, the existing element can be deleted or updated.

### Arrays

There is another type of data objects which can store data in more than two dimensions known as arrays. **"An array is a collection of a similar data type with contiguous memory allocation."** Suppose, if we create an array of dimension (2, 3, 4) then it creates four rectangular matrices of two rows and three columns.

In R, an array is created with the help of **array()** function. This function takes a vector as an input and uses the value in the dim parameter to create an array.

### Matrices

A matrix is an R object in which the elements are arranged in a two-dimensional rectangular layout. In the matrix, elements of the same atomic types are contained. For mathematical calculation, this can use a matrix containing the numeric element. A matrix is created with the help of the matrix() function in R.

Data Frames

A **data frame** is a two-dimensional array-like structure, or we can say it is a table in which each column contains the value of one variable, and row contains the set of value from each column.

There are the following characteristics of a data frame:

1. The column name will be non-empty.
2. The row names will be unique.
3. A data frame stored numeric, factor or character type data.
4. Each column will contain same number of data items.

Factors

**Factors** are also data objects that are used to categorize the data and store it as levels. Factors can store both strings and integers. Columns have a limited number of unique values so that factors are very useful in columns. It is very useful in data analysis for statistical modeling.

Factors are created with the help of **factor()** function by taking a vector as an input parameter.

## Object-Oriented Programming in R

Object oriented programming is a programming model that revolves around classes and objects rather than functions. In R we can also create two special types of classes, S3 and S4. The OOPS concept allows us to create modular pieces that act as the building blocks of the large systems. The S3 and S4 are the two important classes in object oriented programming. The S3 class allows overloading any function. The S4 class differs slightly from S3 as it contains helper functions for defining methods and generics.

### Classes and Objects in R

A class is a user-defined data type from which objects are created. They are also referred to as blueprints sometimes. This is because they define the structure of objects. A class is a passive entity whereas an object is an active entity.

R considers all things as objects. An **object** is a single instance of a class. R provides us a class() function which can be either used to define the class of a function or get the information of the class for an already defined object.

A **class** in R is present in a vector form. Due to this property, the objects can inherit from many classes, and also we can specify the order of inheritance even for complex classes.

## Classes in R

### S3 Class

An S3 class is one of the most used classes in R. This class is easy to implement and many predefined classes are of this type.

The object of an S3 class is a list having class attributes assigned with some names. The component of the list is represented by the member variable of the object.

### Creating S3 Class

We can create an S3 object by the following steps −

* Make a list having the required component values.
* Then, we can create a class easily using the class() function and also append a name to this class.
* We have generic functions in R that serve the purpose of **polymorphism** (like many other oops languages: C++, Java, etc).
* **Polymorphism** is made of two words − **poly** and **morphism** where **poly** means many and **morphism** means forms. It states that a single message can have more than one different form or it can serve different purposes.
* For example in R, the summary() function is a set of many summary() functions that can be used for different data types and data structures. Such functions invoke an adequate function depending upon the type of object passed as an argument. We can see the various implementations of summary() functions using the methods() function.
* methods(summary)
* We can create our own generic function in R. Let us consider the following program that creates a list by the name “myList” and then creates a class out of it.

### Attributes

The **attribute** is a piece of extra information associated with an object and used to handle the object. We can use the **attributes()** function to get the attributes of an object.

### Inheritance in S3 Class

**Inheritance** is one of the most important pillars of object-oriented programming language which allow one class to inherit the properties of another class. This feature leads to benefits like code-reusability.

The S3 class doesn’t contain any particular pattern for its definition.

### S4 Class

R allows us to use another peculiar type of class known as the “S4” class. This class contains the predefined definition. This class contains functions to define methods and generics. This class also provides us with auxiliary functions to define generics and methods.

The setClass() function creates an S4 class. R provides us the new() function which is used to create an object of the S4 class −

### Inheritance in S4 class

Inheritance is one of the most important pillars of object-oriented programming language which allow one class to inherit the properties of another class. This feature leads to benefits like code-reusability.

The S4 class contains a proper pattern for its definition. The derived classes are capable enough to inherit attributes and methods from the base class. In order to achieve this we can