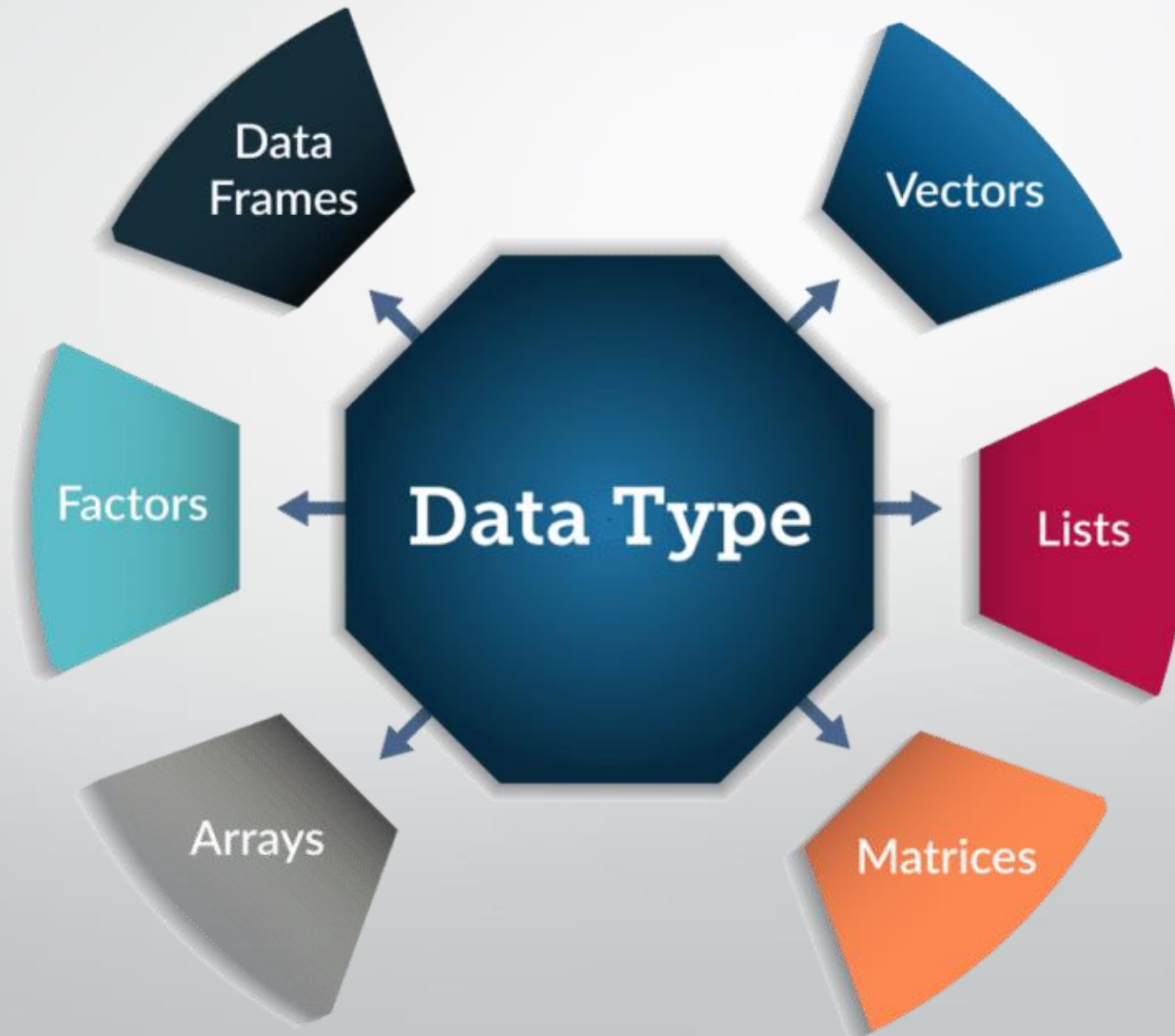




Big Data Analytics

R data structure



Vectors

- In R programming, the very basic data types are the R-objects called **vectors**
- When you want to create vector with more than one element, you should use **c()** function which means to combine the elements into a vector.



Vectors

Console

Terminal x

~/

```
> # 1- Create a vector.  
> apple <- c('red','green',"yellow")  
> print(apple)  
[1] "red"      "green"    "yellow"  
> # Get the class of the vector.  
> print(class(apple))  
[1] "character"  
> |
```

Lists

- A list is an R-object which can contain many different types of elements inside it like vectors, functions and even another list inside it.

```
Console Terminal x
~/
> # 2- Create a list.
> list1 <- list(c(2,5,3),21.3,"red")
> # Print the list.
> print(list1)
[[1]]
[1] 2 5 3

[[2]]
[1] 21.3

[[3]]
[1] "red"

> |
```

Matrices

- A matrix is a two-dimensional rectangular data set. It can be created using a vector input to the matrix function.

```
Console Terminal x
~/
> # 3- Create a matrix.
> M = matrix( c('a','a','b','c','b','a'), nrow=2,ncol=3,byrow = TRUE)
> print(M)
      [,1] [,2] [,3]
[1,] "a"  "a"  "b"
[2,] "c"  "b"  "a"
> |
```

Arrays

- While matrices are confined to two dimensions,
- arrays can be of any number of dimensions.
- The array function takes a dim attribute which creates the required number of dimension.
- In the below example we create an array with two elements which are 3x3 matrices each.

Arrays

Console

Terminal x

~/

```
> # 4- Create an array.  
> a <- array(c('green','yellow'),dim=c(3,3,2))  
> print(a)  
, , 1  
  
      [,1]      [,2]      [,3]  
[1,] "green" "yellow" "green"  
[2,] "yellow" "green"  "yellow"  
[3,] "green"  "yellow" "green"  
  
, , 2  
  
      [,1]      [,2]      [,3]  
[1,] "yellow" "green"  "yellow"  
[2,] "green"  "yellow" "green"  
[3,] "yellow" "green"  "yellow"  
  
>
```


Factors

- Factors are the r-objects which are created using a vector.
- It stores the vector along with the distinct values of the elements in the vector as labels.
- The labels are always character irrespective of whether it is numeric or character or Boolean etc.
- in the input vector. They are useful in statistical modeling.
- Factors are created using the **factor()** function. The **nlevels** functions gives the count of levels.

Factors

Console

Terminal x

~/

```
> # 5- Create a vector.  
> apple_colors <- c('green','green','yellow','red','red','red','green')  
> # Create a factor object.  
> factor_apple <- factor(apple_colors)  
> # Print the factor.  
> print(factor_apple)  
[1] green  green  yellow red    red    red    green  
Levels: green red yellow  
> print(nlevels(factor_apple))  
[1] 3  
> |
```

Data Frames

- Data frames are tabular data objects.
- Unlike a matrix in data frame each column can contain different modes of data.
- The first column can be numeric while the second column can be character and third column can be logical.
- It is a list of vectors of equal length.
- Data Frames are created using the **data.frame()** function.

Data Frames

Console

Terminal x

~/

```
> # 6- Create the data frame.  
> BMI <- data.frame(  
+ gender = c("Male", "Male", "Female"),  
+ height = c(152, 171.5, 165),  
+ weight = c(81, 93, 78),  
+ Age = c(42, 38, 26)  
+ )  
> print(BMI)  
  gender height weight Age  
1   Male  152.0     81  42  
2   Male  171.5     93  38  
3 Female  165.0     78  26  
> |
```