**For Loop in R with Examples for List and Matrix**

A for loop is very valuable when we need to iterate over a list of elements or a range of numbers. Loop can be used to iterate over a list, data frame, vector, matrix or any other object. The braces and square bracket are compulsory.

In this tutorial, we will learn,

* [For Loop Syntax and Examples](https://www.guru99.com/r-for-loop.html" \l "1)
* [For Loop over a list](https://www.guru99.com/r-for-loop.html" \l "2)
* [For Loop over a matrix](https://www.guru99.com/r-for-loop.html" \l "3)

**For Loop Syntax and Examples**

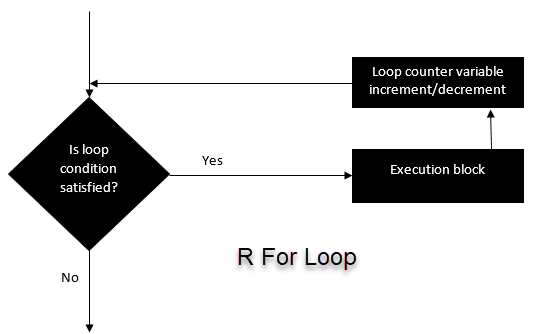
For (i in vector) {

Exp

}

Here,

R will loop over all the variables in vector and do the computation written inside the exp.

[](https://www.guru99.com/images/r_programming/032818_1243_ForLoopinRw1.png)

Let's see a few examples.

**Example 1**: We iterate over all the elements of a vector and print the current value.

# Create fruit vector

fruit <- c('Apple', 'Orange', 'Passion fruit', 'Banana')

# Create the for statement

for ( i in fruit){

print(i)

}

**Output:**

## [1] "Apple"

## [1] "Orange"

## [1] "Passion fruit"

## [1] "Banana"

**Example 2**: creates a non-linear function by using the polynomial of x between 1 and 4 and we store it in a list

# Create an empty list

list <- c()

# Create a for statement to populate the list

for (i in seq(1, 4, by=1)) {

list[[i]] <- i\*i

}

print(list)

**Output:**

## [1] 1 4 9 16

The for loop is very valuable for machine learning tasks. After we have trained a model, we need to regularize the model to avoid over-fitting. Regularization is a very tedious task because we need to find the value that minimizes the loss function. To help us detect those values, we can make use of a for loop to iterate over a range of values and define the best candidate.

**For Loop over a list**

Looping over a list is just as easy and convenient as looping over a vector. Let's see an example

# Create a list with three vectors

fruit <- list(Basket = c('Apple', 'Orange', 'Passion fruit', 'Banana'),

Money = c(10, 12, 15), purchase = FALSE)

for (p in fruit)

{

print(p)

}

**Output:**

## [1] "Apple" "Orange" "Passion fruit" "Banana"

## [1] 10 12 15

## [1] FALSE

**For Loop over a matrix**

A matrix has 2-dimension, rows and columns. To iterate over a matrix, we have to define two for loop, namely one for the rows and another for the column.

# Create a matrix

mat <- matrix(data = seq(10, 20, by=1), nrow = 6, ncol =2)

# Create the loop with r and c to iterate over the matrix

for (r in 1:nrow(mat))

for (c in 1:ncol(mat))

print(paste("Row", r, "and column",c, "have values of", mat[r,c]))

**Output:**

## [1] "Row 1 and column 1 have values of 10"

## [1] "Row 1 and column 2 have values of 16"

## [1] "Row 2 and column 1 have values of 11"

## [1] "Row 2 and column 2 have values of 17"

## [1] "Row 3 and column 1 have values of 12"

## [1] "Row 3 and column 2 have values of 18"

## [1] "Row 4 and column 1 have values of 13"

## [1] "Row 4 and column 2 have values of 19"

## [1] "Row 5 and column 1 have values of 14"

## [1] "Row 5 and column 2 have values of 20"

## [1] "Row 6 and column 1 have values of 15"

## [1] "Row 6 and column 2 have values of 10"