

Loop and Control flow

Neeraj Jain

17-02-2020

1. Loop and control flow function (if)

1. Loop

```
#Creating simple Loop in R  
#Example: 1 - Print  
for(i in 1:9) {  
  print(i^2)  
}
```

```
## [1] 1  
## [1] 4  
## [1] 9  
## [1] 16  
## [1] 25  
## [1] 36  
## [1] 49  
## [1] 64  
## [1] 81
```

#For each i, entire body of the loop will be evaluated in a sequence.

```
#Example: 2 - Mean of each column without apply  
mat <- matrix(1:16, ncol = 4)  
  
apply(mat, 2, mean) #with apply
```

```
## [1] 2.5 6.5 10.5 14.5
```

```
#Manually (one by one)  
mean(mat[, 1])
```

```
## [1] 2.5
```

```
mean(mat[, 2])
```

```
## [1] 6.5
```

```
mean(mat[, 3])
```

```
## [1] 10.5
```

```
mean(mat[, 4])
```

```
## [1] 14.5
```

```

#using loop
a <- c()
for(i in 1:4) {
  a[i] <- mean(mat[, i])
}

#Example: 3 - aggregate using Loop
#Generating some data
set.seed(123)
n <- c("Rishu", "Rajat", "Udit", "Dhawal", "Manan")
Df <- data.frame(Names = sample(x = n, size = 30, replace = TRUE),
                 Marks = rnorm(n = 30, mean = 10, sd = 3))
head(Df)

##   Names      Marks
## 1  Udit   6.921987
## 2  Udit   7.813326
## 3 Rajat   8.124882
## 4 Rajat   4.939920
## 5  Udit  12.513361
## 6 Manan  10.460119

str(Df)

## 'data.frame':   30 obs. of  2 variables:
##  $ Names: Factor w/ 5 levels "Dhawal","Manan",...: 5 5 3 3 5 2 1 4 3 5 ...
##  $ Marks: num   6.92 7.81 8.12 4.94 12.51 ...

#Requirement: using loop, get mean marks and max marks of each name in `n`.
#Manually (for one student)
mean(Df[Df$Names == "Dhawal", 2])

## [1] 8.656977

max(Df[Df$Names == "Dhawal", 2])

## [1] 11.66175

# max(Df[Df$Names == "Dhawal", ]$Marks) #another way of subsetting

#using loop
Mean <- c() #creating empty mean vector
Max <- c() #creating empty max vector
for(i in 1:length(n)) {
  D <- Df[Df$Names == n[i], ]$Marks
  Mean[i] <- mean(D)
  Max[i] <- max(D)
}

(Summary <- data.frame(Names = n, Mean1 = Mean, Max1 = Max))

##   Names      Mean1      Max1
## 1  Rishu  11.670326  16.50687
## 2  Rajat   8.447368  11.27939
## 3   Udit  10.178045  13.62389
## 4 Dhawal   8.656977  11.66175
## 5  Manan  10.851668  12.68538

```

Loop Exercise: Plot using Loop

```
#generating Data
set.seed(123)
Matrix <- matrix(data = rnorm(100), ncol = 10)
colnames(Matrix) <- LETTERS[1:10]
```

Assume first column is dependent variable, and all other variables are independent variable. Do:

1. Plot each independent variable against dependent variable.
2. Save all plot in specific directory. Try to use names such that it convey information about the variable.
3. Title in the plot should be "Scatterplot: A ~ name of IV"

2. Control Flow: if function

The function has major two part. The first is condition `if(condition)`, and second is body (usually written in curly bracket). The expression in the body of `if` function is evaluated only if the condition in `if` is `TRUE`. If condition is not true, then the expression in `if` body is not evaluated at all. Take very simple example:

```
if(TRUE) {
  a <- 2
}
```

```
## [1] 2
```

```
if(FALSE) {
  b <- 2
}
```

```
b
```

```
## Error in eval(expr, envir, enclos): object 'b' not found
```

Another example is square root.

```
#Square root (if number is positive)
srt <- function(x) {
  if(x > 0) {
    x^0.5
  } else {
    print("Number is negative")
  }
}
```

```
#example:
srt(10)
```

```
## [1] 3.162278
```

```
srt(-2)
```

```
## [1] "Number is negative"
```

```
#Annuity Function
Annuity <- function(A = 1, i, n) {
  if(i == 0) {
    n*A
```

```

    } else {
      A * (1 - (1+i)^(-n) ) / i
    }
  }
}

```

#example

```
Annuity(i = .10, n = 10)
```

```
## [1] 6.144567
```

```
Annuity(i = 0 , n = 10)
```

```
## [1] 10
```

#Annuity Function for different payments

```

Annuity <- function(A = 1, i, n, type = c("Monthly","HY", "Q", "A")) {
  if(n < 0) {
    stop("N is negative")
  }
  Fac <- c(Monthly = 12, HY = 2, Q = 4, A = 1)
  s <- Fac[type]
  if(i == 0) {
    n*A*s
  } else {
    A * (1 - (1+i/s)^(-n*s) ) / (i/s)
  }
}

```

#Example

```
Annuity(10000, i = 0, n = 10, type = "Q")
```

```
##      Q
```

```
## 4e+05
```

```
Annuity(100, i = .10, n = 5, type = "Monthly")
```

```
## Monthly
```

```
## 4706.537
```

GitHub Page

Check my Github page for all repositories: github.com/neeraj2308