

R Notebook

Reciprocal of Vector

```
vect <- c(1,2,3,4,5,6,7,8) #creating a vector

reci <- 1/vect          #creating a other vector of reciprocals of each value of the existing vector
print(reci)
```

```
## [1] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571
## [8] 0.1250000
```

```
v <- c(1,2,3,4,5)
u <- c(2,3,4,5,6)

print(v/u)          #floating point quotient of two vectors element wise
```

```
## [1] 0.5000000 0.6666667 0.7500000 0.8000000 0.8333333
```

```
print(v%u)          #Remainder of two vectors element wise
```

```
## [1] 1 2 3 4 5
```

```
print(v%/%u)        # it gives the integer quotient
```

```
## [1] 0 0 0 0 0
```

```
print(u+v)          #addition two vectors
```

```
## [1] 3 5 7 9 11
```

```
print(u*v)          # multiplication of 2 vectors element wise
```

```
## [1] 2 6 12 20 30
```

```
v <- LETTERS[1:6]    #creating a list of values 1,2,3,4,5,6
for ( i in v)         #traversing through list
{
  if (i == 'D')        #condition
  { next }
  print(i)             #printing the list of values
}
```

```
## [1] "A"
## [1] "B"
## [1] "C"
## [1] "E"
## [1] "F"
```

```
employees <- list('jane', 'john','pavan','pyla','nothing') #creating the list
for (employee in employees) #traversing through list and printing the values
{
  print(employee)
}
```

```
## [1] "jane"
## [1] "john"
## [1] "pavan"
## [1] "pyla"
## [1] "nothing"
```

```
a <- 8
b <- 12
c <- 1:10
print(a %in% c) #checking wheather '8' in list C or not
```

```
## [1] TRUE
```

```
print(b %in% c) #checking wheather '12' in list C or not
```

```
## [1] FALSE
```

```
M = matrix( c(1,1,0,0), nrow=2,ncol=2,byrow = TRUE)
t = M %*% t(M) # t(M) will give the transpose of the matrix
print(t)
```

```
##      [,1] [,2]
## [1,]    2    0
## [2,]    0    0
```

Control Structures

IF statement

```
x <-8
if(is.integer(x)) #if statement
{
  print("X is an Integer") #executing the if statement
}
```

```

if (2 > 0)                #if statement
{
  print("true")           #printing the value if IF statement is true
}

```

```
## [1] "true"
```

```

x <- 100

if(x > 10){
  print(paste(x, "is greater than 10"))
}

```

```
## [1] "100 is greater than 10"
```

IF then ELSE statements

```

x <- c(8, 3, -2, 5)

# without curly braces
if(any(x < 0)) print("x contains negative numbers")

```

```
## [1] "x contains negative numbers"
```

```

## [1] "x contains negative numbers"

# with curly braces produces same result
if(any(x < 0)){
  print("x contains negative numbers")
}

```

```
## [1] "x contains negative numbers"
```

```

## [1] "x contains negative numbers"

# an if statement in which the test expression is FALSE
# does not produce any output
y <- c(8, 3, 2, 5)

if(any(y < 0)){
  print("y contains negative numbers")
}

```

```

x <- c("what", "is", "truth")
if("Truth" %in% x)
{
  print("Truth is found")
} else
{
  print("Truth is not found")
}

```

```
## [1] "Truth is not found"
```

```
x <- c("what","is","truth")
if("Truth" %in% x)
{
  print("Truth is found the first time")
}else if("truth" %in% x)
{
  print("truth is found the second time")      # if else if else statements
}else
{
  print("No truth found")
}
```

```
## [1] "truth is found the second time"
```

```
x <- 5

# Check value is less than or greater than 10
if(x > 10){
  print(paste(x, "is greater than 10"))
}else{
  print(paste(x, "is less than 10"))           #printing according to condition
}
```

```
## [1] "5 is less than 10"
```

Loops

For Loop

```
x <- letters[4:10]

for(i in x){
  print(i)                                     #for Loop implemetation
}
```

```
## [1] "d"
## [1] "e"
## [1] "f"
## [1] "g"
## [1] "h"
## [1] "i"
## [1] "j"
```

```
# Defining matrix
m <- matrix(2:15, 2)

for (r in seq(nrow(m))) {
  for (c in seq(ncol(m))) {                  #Matrix traversal using for loop

```

```

    print(m[r, c])
  }
}

```

```

## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
## [1] 14
## [1] 3
## [1] 5
## [1] 7
## [1] 9
## [1] 11
## [1] 13
## [1] 15

```

```

v <- LETTERS[1:4]
for ( i in v)                # #for Loop implemetation
{
  print(i)
}

```

```

## [1] "A"
## [1] "B"
## [1] "C"
## [1] "D"

```

While Loop

```

x = 1

# Print 1 to 5
while(x <= 5)
{
  #While Implementation
  print(x)
  x = x + 1
}

```

```

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5

```

```

i <- 1
# While i is less than or equal to three, print i
# The loop will increment the value of i after each print

```

```
while (i <= 3)
{
  print(i)
  i <- i + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
```

R Repeat Loop

```
x = 1

# Print 1 to 5
repeat{
  print(x)
  x = x + 1
  if(x > 5){
    break
  }
}
```

#Repeat loop in R implementation

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```
employees <- list("jane", "john")
for (employee in employees)
{
  print(employee)
}
```

```
## [1] "jane"
## [1] "john"
```

```
x <- 20
if (x < 20)
{ print("x is less than 20") } else if (x > 20)
{ print("x is greater than 20") } else { print("x is equal to 20") }
```

```
## [1] "x is equal to 20"
```

```
v <- c("Om","Sri","Sairam")
count <- 1
repeat{
  print(v)
  count <- count+1
}
```

#printing the vector until loop breaks

```

if(count > 5)
{ break }
}

```

```

## [1] "Om"      "Sri"      "Sairam"
## [1] "Om"      "Sri"      "Sairam"
## [1] "Om"      "Sri"      "Sairam"
## [1] "Om"      "Sri"      "Sairam"
## [1] "Om"      "Sri"      "Sairam"

```

Loop Control Statements

Break

```

v <- LETTERS[1:6]
for ( i in v)
{
  if (i == "D")
  { break }                      #Putting break to the loop if D appear in list
  print(i)
}

```

```

## [1] "A"
## [1] "B"
## [1] "C"

```

```

                                # Defining vector
x <- 1:10

                                # Print even numbers
for(i in x){
  if(i%%2 != 0){
    break                      #breaking from the loop
  }
  print(i)
}

```

Next Statement

```

# Defining vector
x <- 1:10

# Print even numbers
for(i in x){
  if(i%%2 != 0){
    next #Jumps to next loop
  }
  print(i)
}

```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
```