$MDSC - 201(P)$

$ASSIGNMENT - II$

Author : Akula Venkata Satya Sai Gopinadh

Date: 11-02-2023

Arithmetic Operations between Two Vectors.

Let’s start one by one in an order.

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

c <- a+b #Adding Two Vectors
print("Addition of two vectors is: ")
```

```
## [1] "Addition of two vectors is: "
```

```
c
```

```
## [1]  3 12 15 15
```

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

d <- a-b # Subtracting second vector from the first vector
print("Subtracting of two vectors is: ")
```

```
## [1] "Subtracting of two vectors is: "
```

```
d
```

```
## [1] -1  4  3 -1
```

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

e <- a*b # Multiplying two vectors
print("Multiplication  of two vectors is: ")
```

```
## [1] "Multiplication  of two vectors is: "
```

```
e
```

```
## [1]  2 32 54 56
```

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

f <- a/b # Dividing the first vector with second
print("Division of two vectors is: ")
```

```
## [1] "Division of two vectors is: "
```

```
f
```

```
## [1] 0.500 2.000 1.500 0.875
```

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

g <- a%%b # Gives the remainder of first vector with second
print("Remainder of first vector with the second vector : ")
```

```
## [1] "Remainder of first vector with the second vector : "
```

```
g
```

```
## [1] 1 0 3 7
```

```
a<- c(1,8,9,7)
b<- c(2,4,6,8)

h <- a%/%b # Gives the quotient
print("Quotient of the first over the second vector is: ")
```

```
## [1] "Quotient of the first over the second vector is: "
```

```
h
```

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

Special Operations

```
# (:) Colon Operation

i <- 25:45
print("Gives numbers from 25 to 45 ")
```

```
## [1] "Gives numbers from 25 to 45 "
```

```
i
```

```
##  [1] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
```

```
 # %in%    This operator is used to identify if an element belongs to a vector.

a <- 2
b <- 1
c <- 2:3

print(a %in% c)
```

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

```
print(b %in% c)
```

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

```
# %*%   This operator is used to multiply a matrix with its transpose

#example 1
print("EXAMPLE-1")
```

```
## [1] "EXAMPLE-1"
```

```
M = matrix( c(1,2,3,4,5,6,7,8,9), nrow=3,ncol=3,byrow = TRUE)
print("Given Matrix : ")
```

```
## [1] "Given Matrix : "
```

```
print(M)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

```
t = M %*% t(M)
print("After Multiplying with its transpose")
```

```
## [1] "After Multiplying with its transpose"
```

```
print(t)
```

```
##      [,1] [,2] [,3]
## [1,]   14   32   50
## [2,]   32   77  122
## [3,]   50  122  194
```

```
#example 2
print("EXAMPLE-2")
```

```
## [1] "EXAMPLE-2"
```

```
N = matrix( c(1,2,2,5), nrow=2,ncol=2,byrow = FALSE)
print("Given Matrix : ")
```

```
## [1] "Given Matrix : "
```

```
print(N)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    2    5
```

```
t = N %*% t(N)
print("After Multiplying with its transpose")
```

```
## [1] "After Multiplying with its transpose"
```

```
print(t)
```

```
##      [,1] [,2]
## [1,]    5   12
## [2,]   12   29
```

Control Structures

Control Structures are nothing but they allow us to put some “condition”(logic) in the code.Then only if that condition satisfies only then it will execute that code or else it will go to the next block of code.

We have Commonly used Control Structures :

They are

- **if and else** : testing a condition and acting on it.
- **for** : execute a loop a fixed number of times.
- **while** : execute a loop *while* a condition is true.
- **repeat :** execute an infinite loop (must “break” out of it to stop).
- **break** : break the execution of a loop.
- **next** : skip an interation of a loop.

1. If - Else:

This control structure is used if we have a two code to execute and if the condition is met then it will execute one code, else it will execute the other code.

```
j <- 6

if (j > 10){
  print("j is greater than 10")
}else{
  print("j is not greater than 10")
}
```

```
## [1] "j is not greater than 10"
```

2.If-else-if :

#Example-1

```
k <- 2.5

if (k > 5){
  print("k is greater than 5")
}else if(k < 1){
  print("k is less than 1")
}else{
  print("k is between 1 and 5")
}
```

```
## [1] "k is between 1 and 5"
```

#Example-2

Taking USER INPUT :

```
Name = readline(prompt = "Enter your Name : ")
```

```
## Enter your Name :
```

```
print(Name)
```

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

```
age = readline(prompt = 'Enter your Age : ')
```

```
## Enter your Age :
```

```
print(age)
```

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

```
if (age > 18){
  print("Your are eligible to vote")
}else {
  print("You need to grow more kid!! to cast your vote ")
}
```

```
## [1] "You need to grow more kid!! to cast your vote "
```

3.For Loop :

#Example-1:

```
for (i in 1:20){
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
```

#Example-2:

```
# lets add all the values of a vector using for loop

m <- c(20,45,17.5,12.5,5,8.0)

sum <- 0

for(i in m ){
  sum <- sum + i
}
a <- sum
print(paste("Sum of the elements of the vector is :",a))
```

```
## [1] "Sum of the elements of the vector is : 108"
```

#Example-3:

```
student <- list("Aditya",'Kumar',"Satya","Vamsee","Hemanth","Lalith",
                "Pyla","Mathura","Saideva","Sai Kumar","Aharniesh","Anirudha",
                "Srihari","Swaroop")

for (i in 1:length(student)) {
  print(student[[i]])
}
```

```
## [1] "Aditya"
## [1] "Kumar"
## [1] "Satya"
## [1] "Vamsee"
## [1] "Hemanth"
## [1] "Lalith"
## [1] "Pyla"
## [1] "Mathura"
## [1] "Saideva"
## [1] "Sai Kumar"
## [1] "Aharniesh"
## [1] "Anirudha"
## [1] "Srihari"
## [1] "Swaroop"
```

4.While loop :

While loops begin by testing a condition. If it is true, then they execute the loop body. Once the loop body is executed, the condition is tested again, and again , until the condition is false, after which the loop exits.

```
count <- 0

while(count < 15) {
  print(count)
  count<- count+1
}
```

```
## [1] 0
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
```

```
count <- 0

while(count < 15) {

  count<- count+1
}
 print(count)
```

```
## [1] 15
```

5. Repeat loop :

```
n <- c("Hi , Myself Satya !")
i <- 1
repeat {

   print(n)
   i <- i + 1
   if(i >5) {
      break
   }
}
```

```
## [1] "Hi , Myself Satya !"
## [1] "Hi , Myself Satya !"
## [1] "Hi , Myself Satya !"
## [1] "Hi , Myself Satya !"
## [1] "Hi , Myself Satya !"
```

```
o <- 1
i <- 1

repeat {
print(o)

i <- i + 1
o = o + 1

if(i > 25) {
    break
}
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
```

6.Break :

This will terminate a looping condition.

```
for (i in 1:20){ # this will print numbers from 1 to 7 and terminates as it reaches 8
  if (i==8){
    break
  }
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
```

7.Next :

This NEXT is used to skip an iteration in any loop like while and for.

```
#Let's see an example of print only even numbers from 1 to 20

for(i in 1:20){
  if (i %% 2 != 0){
    next
  }
  print(i)
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
## [1] 14
## [1] 16
## [1] 18
## [1] 20
```

SCATTER PLOT USING R DATASETS:

```
data("mtcars")# loading dataset
```

```
a <- nrow(mtcars)
print(paste("Rows in mtcars dataset : ",a))
```

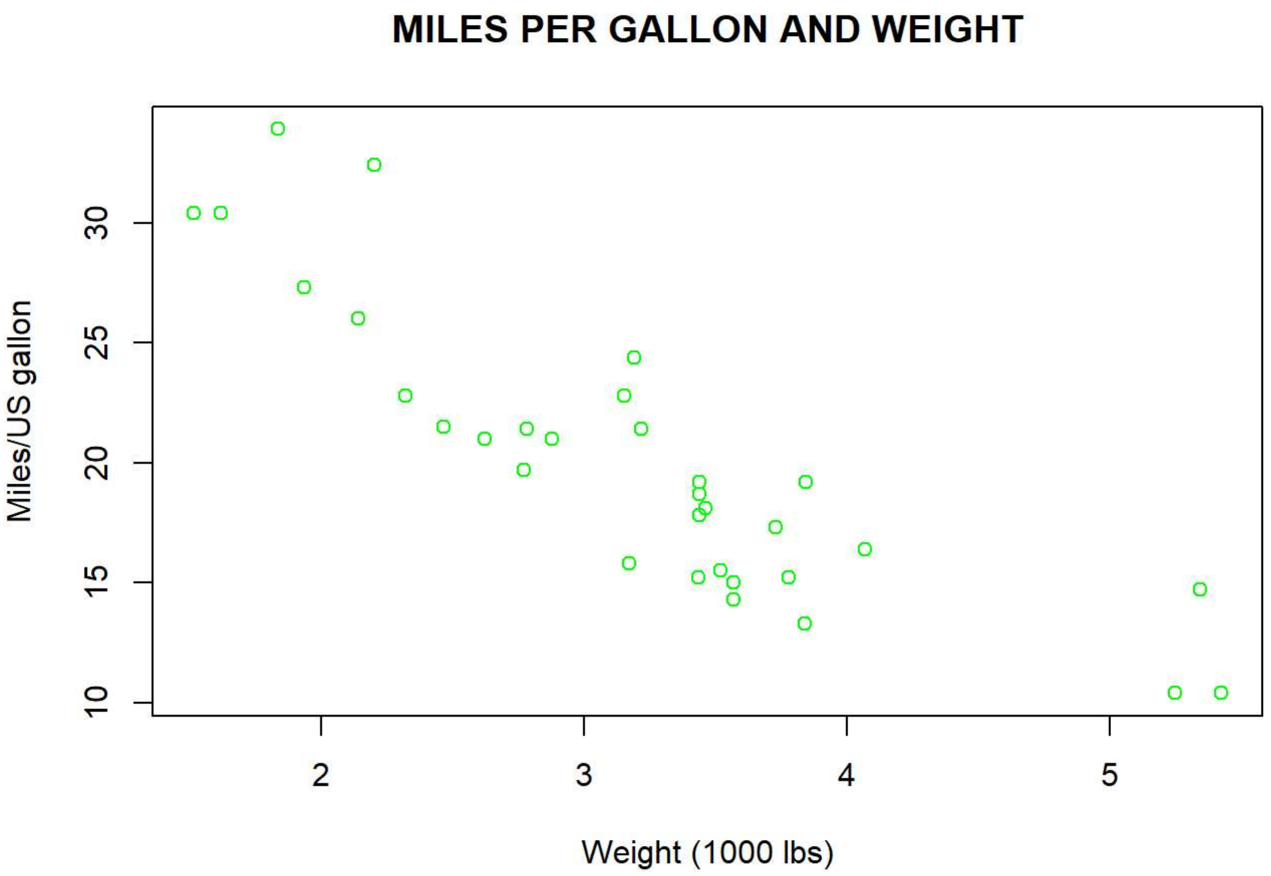
```
## [1] "Rows in mtcars dataset :  32"
```

```
b <- ncol(mtcars)
print(paste("Columns in mtcars dataset : ",b))
```

```
## [1] "Columns in mtcars dataset :  11"
```

```
attach(mtcars)
```

```
plot(wt,mpg,col="green",ylab='Miles/US gallon',xlab ='Weight (1000 lbs)',
     main= "MILES PER GALLON AND WEIGHT")
```


MILES PER GALLON AND WEIGHT
Miles/US gallon
Weight (1000 lbs)