#Variable Assignment – (2 ways to assign the variable)

X = 2

X

[1] 2

Y<-2

Y

[1] 2

#Data Types

a = 2.5 #numeric

A

[1] 2.5

class(a)

[1] "numeric"

b=1 #also Numeric

B

[1] 1

class(b)

[1] "numeric"

c=1L #integer

c[1] 1

class(c)[1] "numeric"

d <- "Hello"

print(d)[1] 1

class(d)[1] "integer"

e=TRUE #logical (TRUE, FALSE)

e[1] TRUE

class(e)[1] "logical"

f=1+4i # comple

f[1] 1+4i

class(f)[1] "complex"

#Arithmetic operators +, -, \*, /, ^, or \*\*, %%, (%/%)

1+1 [1] 2

2-2[1] 0

2\*2[1] 4

2\*\*2[1] 4

6/3[1] 2

2^2[1] 4

4%%2 #(modulus) (you will get the reminder part)

5%%2[1] 0

15%%3 [1] 1

#Relational operators || Logical operator(>, >=, <, <=,==,!=)

5>3[1] TRUE

2<3[1] TRUE

5>=5[1] TRUE

5>5[1] FALSE

10==10[1] TRUE

10!=5[1] TRUE

A<-110

a<-10

#Sequence function

#1 – possibility

seq1 = 1:10

seq1 [1] 1 2 3 4 5 6 7 8 9 10

seq\_len(10) [1] 1 2 3 4 5 6 7 8 9 10

#2 – possibility

seq2 = seq(1,10)

seq2[1] 1 2 3 4 5 6 7 8 9 10

seq3 = seq(1,10,by=2)

seq3 [1] 1 3 5 7 9

#DATA STRUCTURES

#Vectors

num\_vec = c(1,2,3) # To create a numeric vector

num\_vec[1] 1 2 3

class(num\_vec)[1] "numeric"

int\_vec = c(1L,2L,3L)

int\_vec[1] 1 2 3

class(int\_vec)[1] "integer"

bool\_vec = c(T,F,T)

bool\_vec[1] TRUE FALSE TRUE

class(bool\_vec)[1] "character"

char\_vec = c("Hello", "World")

char\_vec

class(char\_vec)[1] "character"

mix\_vec = c(T,2L,2)#we assigned as bool,int,numeric value & R is transferring that value to numeric

mix\_vec[1] 1 2 2

class(mix\_vec)[1] "numeric"

mix\_vec\_1 = c(T,2L,2,"bla")

mix\_vec\_1[1] "TRUE" "2" "2" "bla"

class(mix\_vec\_1)[1] "character"

#Matrices

m = matrix(1:4,ncol=2, nrow=2)

M

[,1] [,2]

[1,] 1 3

[2,] 2 4

m1 = matrix(1:6, ncol=2, nrow=2)

M1

[,1] [,2]

[1,] 1 3

[2,] 2 4

m2 = matrix(1:6, ncol=3, nrow=3)

M2

[,1] [,2] [,3]

[1,] 1 4 1

[2,] 2 5 2

[3,] 3 6 3

col\_vec = matrix(1:4, ncol=1,nrow=4)

Col\_vec

[,1]

[1,] 1

[2,] 2

[3,] 3

[4,] 4

## You cannot store different data type in the matrices

#Indexing

char\_vec = c("tick", "tack", "toe")

cbind(char\_vec,bool\_vec)

char\_vec bool\_vec

[1,] "tick" "TRUE"

[2,] "tack" "FALSE"

[3,] "toe" "TRUE"

char\_vec[1][1] "tick"

char\_vec[2][1] "tack"

char\_vec[3][1] "toe"

char\_vec[1:3][1] "tick" "tack" "toe"

char\_vec[1:5][1] "tick" "tack" "toe" NA NA

char\_vec[3:3][1] "toe"

#Indexing from matrix

m2 = matrix(1:9, ncol=3, nrow=3)

M2

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

m2[1,][1] 1 4 7

m2[,1][1] 1 2 3

m2[1,1][1] 1

m2[1,2][1] 4

m2[2,][1] 2 5 8

m2[2:3] [1] 2 3

m2[2,3][1] 8

#LIST (TO STORE MULTIPLE DATA TYPES)

list1 = list(c(1,2,3), 'hello', 1:10)

str(list1) # find the data type in the list

List of 3

$ : num [1:3] 1 2 3

$ : chr "hello"

$ : int [1:10] 1 2 3 4 5 6 7 8 9 10

#DATAFRAME

df = data.frame(

names = c("chancy","dushya"),

debts = c(100,200))

str(df)

'data.frame': 2 obs. of 2 variables:

$ names: chr "chancy" "dushya"

$ debts: num 100 200

df[,2][1] 100 200

Df[1,]

names debts

1 chancy 100

df[1,2][1] 100

df[1,1][1] "chancy"

#Read the csv file

df1= read.csv("D:/NIT/FEBRUARY/7 & 8 feb (praticle tensorflow)/6th, 7th - Deep Learning/Practicle - CPU/ANN\_ 1st/Churn\_Modelling.csv")

View(df1)

mean(df1[,2])[1] 15690941

#for loop

for(value in 1:5){print("hello")}

[1] "hello"

[1] "hello"

[1] "hello"

[1] "hello"

[1] "hello"

vec = c(1,2,3)

for(value in vec) {print(vec)}

[1] 1 2 3

[1] 1 2 3

[1] 1 2 3

#while loop

value = 1

while(value<5) {print(value)

value = value + 1}

-------- while(TRUE) {print('infinity')}

#Function

x = function(){print('hello')}

x()[1] "hello"

y=function(arg1){arg1\*\*2}

y(2)[1] 4

z=function(arg1,arg2){loc = arg1 + arg2

return(loc)}

z(10,20)[1] 30

#recursive function (factorial)

facfoo = function(n) {if (n==0) {return(1)}

else return (facfoo(n-1) \* n)}

facfoo(5)[1] 120

facfoo(4)[1] 24