

FirstProgram.R

RKC

2024-10-25

```
print("Hello World!")
```

```
## [1] "Hello World!"
```

```
x = 2; X=3;Y=4;Z=5;X;Y;Z
```

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

```
## [1] 4
```

```
## [1] 5
```

```
print(x)
```

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

```
class(x)
```

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

```
X = 3;  
print(X)
```

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

```
#### Second Section 0 Assignment ####
```

```
x <- 2  
2 -> x
```

```
x = 2
```

```
z <- 4L  
z
```

```
## [1] 4
```

```
class (z)
```

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

```
Z = "Hello"  
Z
```

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

```
class(Z)
```

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

```
Z = "Hi"  
Z
```

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

```
class(Z)
```

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

```
#### Vectors ####
```

```
v = 8:17  
print(v)
```

```
## [1] 8 9 10 11 12 13 14 15 16 17
```

```
#creating sequence from 5.5 to 17.5
```

```
v = 5.5:17.5  
print(v)
```

```
## [1] 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5 14.5 15.5 16.5 17.5
```

```
#if the final element speciifed does not belong to the sequence  
# then it is discarded
```

```
v = 3.8:11.4  
print(v)
```

```
## [1] 3.8 4.8 5.8 6.8 7.8 8.8 9.8 10.8
```

```
#v = -12.3:2.4;z
```

```
# Create vector with elements from 5 to 9 incrementing by 0.5  
v = seq(5,9, by=0.5);v
```

```
## [1] 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0
```

```
length(v)
```

```
## [1] 9
```

```
# Create a vector with elements from 10 to 20 incrementing by 0.1  
seq(10,20,by=0.1)
```

```
## [1] 10.0 10.1 10.2 10.3 10.4 10.5 10.6 10.7 10.8 10.9 11.0 11.1 11.2 11.3 11.4  
## [16] 11.5 11.6 11.7 11.8 11.9 12.0 12.1 12.2 12.3 12.4 12.5 12.6 12.7 12.8 12.9  
## [31] 13.0 13.1 13.2 13.3 13.4 13.5 13.6 13.7 13.8 13.9 14.0 14.1 14.2 14.3 14.4  
## [46] 14.5 14.6 14.7 14.8 14.9 15.0 15.1 15.2 15.3 15.4 15.5 15.6 15.7 15.8 15.9  
## [61] 16.0 16.1 16.2 16.3 16.4 16.5 16.6 16.7 16.8 16.9 17.0 17.1 17.2 17.3 17.4  
## [76] 17.5 17.6 17.7 17.8 17.9 18.0 18.1 18.2 18.3 18.4 18.5 18.6 18.7 18.8 18.9  
## [91] 19.0 19.1 19.2 19.3 19.4 19.5 19.6 19.7 19.8 19.9 20.0
```

```
# print even Number between 10 and 20  
seq(10, 20, by = 2)
```

```
## [1] 10 12 14 16 18 20
```

```
# print multiple of 7 from 7 to 100  
seq(7,100, by = 7)
```

```
## [1] 7 14 21 28 35 42 49 56 63 70 77 84 91 98
```

```
### Using c() function  
# numeric data  
v = c(10,15,66,478,99)  
print(v)
```

```
## [1] 10 15 66 478 99
```

```
class(v)
```

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

```
length(v)
```

```
## [1] 5
```

```
# Character data  
b = c("Pranay", "Hemant", "Pratik", "Bhupendra");b
```

```
## [1] "Pranay" "Hemant" "Pratik" "Bhupendra"
```

```
class(b)
```

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

```
c = c("Pranay", "MD AND CEO", 1750000);c
```

```
## [1] "Pranay"      "MD AND CEO" "1750000"
```

```
class(c)
```

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

```
length(c)
```

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

```
# Subsetting
```

```
c[1]
```

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

```
c[5]
```

```
## [1] NA
```

```
c[1:3]
```

```
## [1] "Pranay"      "MD AND CEO" "1750000"
```

```
c[0]
```

```
## character(0)
```

```
# updating elements
```

```
c[2] = "Founder and Board Member"
```

```
c
```

```
## [1] "Pranay"      "Founder and Board Member"
```

```
## [3] "1750000"
```

```
c_2 = c('Pranay', 'Good Morning', 'Howdy Man', 'Im Good')
```

```
c_2
```

```
## [1] "Pranay"      "Good Morning" "Howdy Man"    "Im Good"
```

```

class(c_2)

## [1] "character"

c_2[c(2,3,7)]

## [1] "Good Morning" "Howdy Man"    NA

c_2[1]

## [1] "Pranay"

# How to get characters from prime positions in above c_2

v = c(1:15)

v[c(2,3,5,7,11,13)]

## [1]  2  3  5  7 11 13

# Updating class of vector

r= c('1','2','3','4');class(r)

## [1] "character"

r = as.numeric(r);class(r)

## [1] "numeric"

is.numeric(r)

## [1] TRUE

# Create a factor
grade = c('A','B','C','D','E');grade

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

class(grade)

## [1] "character"

grade = as.factor(grade);class(grade)

## [1] "factor"

```

```
# The non-character values are coerced to character type if one of the elements  
# is character
```

```
s = c('apple', 'red', 'green', TRUE, 'blue', 'magenta', 'brown', 'orange')
```

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"  
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
LETTERS
```

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"  
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
# create a vector which stores first 5 multiples,  
# next 5 multiples of 7 and 5 multiples of 16
```

```
v = c(seq(5,25,5), seq(7,35,7), seq(16,80,16));v
```

```
## [1] 5 10 15 20 25 7 14 21 28 35 16 32 48 64 80
```

```
# consider 3 vectors and append or combine into one
```

```
a = 1:10  
b = c('A', 'B', 'C')  
c = c(TRUE, FALSE)  
  
d = c(a,b,c);d
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"  
## [10] "10" "A" "B" "C" "TRUE" "FALSE"
```

```
?rep
```

```
## starting httpd help server ... done
```

```
rep(10,5)
```

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

```
a = c('Pranay', 'Founder')  
rep(a,15)
```

```
## [1] "Pranay" "Founder" "Pranay" "Founder" "Pranay" "Founder" "Pranay"  
## [8] "Founder" "Pranay" "Founder" "Pranay" "Founder" "Pranay" "Founder"  
## [15] "Pranay" "Founder" "Pranay" "Founder" "Pranay" "Founder" "Pranay"  
## [22] "Founder" "Pranay" "Founder" "Pranay" "Founder" "Pranay" "Founder"  
## [29] "Pranay" "Founder"
```

```

?rev
rev(a)  # Prints reverse

## [1] "Founder" "Pranay"

b = b[-1];b# removes first element from the seq

## [1] "B" "C"

v = -5:5
print(v)

## [1] -5 -4 -3 -2 -1 0 1 2 3 4 5

v[v>0]  #Conditional TRUE elements are brought out

## [1] 1 2 3 4 5

v[v<0]

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

v[v==0]

## [1] 0

v = c(1,2,3)
v[v>0]

## [1] 1 2 3

v[v<0]

## numeric(0)

v[v==0]

## numeric(0)

# Finding index from name
k = c(11,13,15,7,8,9)
which((k==1))

## integer(0)

```

```
which((k==13))
```

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

```
which(k==13)
```

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

```
# creaet a vector "f" store any 50 elements in it and create another vector  
# in "J" and store random 5 positions till 50 in it and then access the elements  
# from f wth specified position in "J"
```

```
F = c(20:50)
```

```
J = c(22,30,40,42)
```

```
F[J] # Brings matched elements of J from F
```

```
## [1] 41 49 NA NA
```

```
J[F]
```

```
## [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  
## [26] NA NA NA NA NA NA
```

```
F[-J] # Brings out all the elemnts from F vector
```

```
## [1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 43 44 45  
## [26] 46 47 48 50
```

```
min(J)
```

```
## [1] 22
```

```
max(J)
```

```
## [1] 42
```

```
d = c(3,3,4,4,5,6,2,4,8,5,1,2,5,6)  
unique(d)
```

```
## [1] 3 4 5 6 2 8 1
```

```
t = c(4,5," ",6,8)  
length(t)
```

```
## [1] 5
```



```
t = c(4,5,NA,6,8)
length(t)
```

```
## [1] 5
```

```
# Numeric Data
num1 = 10.23
class(num1)
```

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

```
# Integer data
num2 = 63L
class(num2)
```

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

```
# Complex Data
num3 = 3+2i
class(num3)
```

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

```
#### Operators ####
```

```
#Arithmetic Operations
```

```
x = 20
y = 3
```

```
x%/%y
```

```
## [1] 6
```

```
x/y
```

```
## [1] 6.666667
```

```
# Modulus
x%%y
```

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

```
#Exponent
x^y
```

```
## [1] 8000
```

```
x**y
```

```
## [1] 8000
```

```
# NonAtomic Vectors
```

```
x_1 = c(1,2,3,4,5)
```

```
y_1 = c(7,8,9,10,11)
```

```
x_1 + y_1
```

```
## [1] 8 10 12 14 16
```

```
x_1 * y_1
```

```
## [1] 7 16 27 40 55
```

```
x_1 - y_1
```

```
## [1] -6 -6 -6 -6 -6
```

```
x_1 / y_1
```

```
## [1] 0.1428571 0.2500000 0.3333333 0.4000000 0.4545455
```

```
x_1 %/% y_1
```

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

```
x_1 ** y_1
```

```
## [1] 1 256 19683 1048576 48828125
```

```
y_1 **x_1
```

```
## [1] 7 64 729 10000 161051
```

```
#
```

```
x_1 = c(1,2,3,4,5,6)
```

```
y_1 = c(5,6,7)
```

```
x_1 %/% y_1
```

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

```
x_1 / y_1
```

```
## [1] 0.2000000 0.3333333 0.4285714 0.8000000 0.8333333 0.8571429
```

```
x_1 + y_1
```

```
## [1] 6 8 10 9 11 13
```

```
x_1 - y_1
```

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

```
x_1 = c(1,2,3,4,5,6,7)
y_1 = c(5,6,7)
x_1 - y_1
```

```
## Warning in x_1 - y_1: longer object length is not a multiple of shorter object
## length
```

```
## [1] -4 -4 -4 -1 -1 -1 2
```

```
## Relational Operator
```

```
x =18
```

```
y = 17
```

```
x<y
```

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

```
x>y
```

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

```
x<=9
```

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

```
x>=9
```

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

```
y == 17
```

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

```
## Relational Operators Vectors
```

```
x_1 = c(17,18,19,20)
```

```
y_1 = c(16,15,14,20)
```

```
x_1 < y_1
```

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

```
x_1 > y_1
```

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

```
x_1 > 8
```

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

```
## Logical Operators
```

```
x = c(TRUE, FALSE, 0, 8)
```

```
y = c(TRUE, FALSE, FALSE, TRUE)
```

```
x
```

```
## [1] 1 0 0 8
```

```
y
```

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

```
!x
```

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

```
x&y
```

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

```
x|y
```

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

```
### Miscellaneous Operators
```

```
## CTRL F (find something)
```

```
# This is used to identify if an element belongs to vector
```

```
v1 = 2
```

```
v2 = 12
```

```
t = 1:10
```

```
print(v1 %in% t)
```

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

```
print(v2 %in% t)
```

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

```
s = 'b'
vowel = c('a','e','i','o','u')
s %in% vowel
```

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

```
# create a vector and replace number lesser than 5 with 0
v = c(6,2,9,1,8,7)
v[v<5]
```

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

```
v[v<5] = 0
v
```

```
## [1] 6 0 9 0 8 7
```

```
# Sort the elements in vector in ascending and desc order
v= c(5,8,2,10,15,3,12)
sort(v)
```

```
## [1] 2 3 5 8 10 12 15
```

```
?sort
sort(v, decreasing = TRUE)
```

```
## [1] 15 12 10 8 5 3 2
```

```
#extract the elements that are in odd position
```

```
b = v[c(1,3,5,7)]
b
```

```
## [1] 5 2 15 12
```

```
# Sum the elements in vector
?sum
sum(v)
```

```
## [1] 55
```

```
#### MATRICES ####
```

```
?matrix
```

```
m = matrix(nrow = 2, ncol = 2);m
```

```
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
```

```
v = c(10,20,30,40)
m = matrix(v)
dim(m)
```

```
## [1] 4 1
```

```
# Create matrix with values from my_vector, 2rows, 2 cols
```

```
m = matrix(v, 2, 2);m
```

```
##      [,1] [,2]
## [1,]   10   30
## [2,]   20   40
```

```
n = matrix(v, 2,2, byrow = T) ;n
```

```
##      [,1] [,2]
## [1,]   10   20
## [2,]   30   40
```

```
m_2 = matrix(1:9, 3, 3, byrow = FALSE);m_2
```

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

```
m_2 = matrix(1:9, 3, 3, byrow = TRUE);m_2
```

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

```
m_2 = matrix(1:6, 3, 3, byrow = TRUE);m_2
```

```
## Warning in matrix(1:6, 3, 3, byrow = TRUE): data length differs from size of
## matrix: [6 != 3 x 3]
```

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

```
m_2 = matrix(1:10, 3, 3, byrow = TRUE);m_2
```

```
## Warning in matrix(1:10, 3, 3, byrow = TRUE): data length [10] is not a
## sub-multiple or multiple of the number of rows [3]
```

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

```
## Accessing values (Slicing)
```

```
m_2[2,2]
```

```
## [1] 5
```

```
m_2[1,2]
```

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

```
m_2[3,1]
```

```
## [1] 7
```

```
m_2[2,]
```

```
## [1] 4 5 6
```

```
m_2[1,]
```

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

```
# Accessing Sub-Matrix
```

```
m_2[1:2, 2:3]
```

```
##      [,1] [,2]
## [1,]    2    3
## [2,]    5    6
```

```
m_2[2:3,3:3]
```

```
## [1] 6 9
```

```
m_2[-3:-3]
```

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

```
## Transpose Matrix
```

```
t(m_2)
```

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

```
# I don't know the number of row how to create the matrix?
```

```
m_2 = matrix(1:26, ncol=3);m_2
```

```
## Warning in matrix(1:26, ncol = 3): data length [26] is not a sub-multiple or
## multiple of the number of rows [9]
```

```
##      [,1] [,2] [,3]
## [1,]    1   10   19
## [2,]    2   11   20
## [3,]    3   12   21
## [4,]    4   13   22
## [5,]    5   14   23
## [6,]    6   15   24
## [7,]    7   16   25
## [8,]    8   17   26
## [9,]    9   18    1
```

```
m_2 = matrix(1:26, 3);m_2
```

```
## Warning in matrix(1:26, 3): data length [26] is not a sub-multiple or multiple
## of the number of rows [3]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    4    7   10   13   16   19   22   25
## [2,]    2    5    8   11   14   17   20   23   26
## [3,]    3    6    9   12   15   18   21   24    1
```

```
nrow(m_2)# to get number of rows in matrix
```

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

```
ncol(m_2)# to get number of cols in matrix
```

```
## [1] 9
```

```
# Additions rowwise
```

```
rowSums(m_2)
```

```
## [1] 117 126 109
```



```
# Additions columnwise  
colSums(m_2)
```

```
## [1]  6 15 24 33 42 51 60 69 52
```

```
# Diagonal Matrix
```

```
d = diag(2,3,3);d
```

```
##      [,1] [,2] [,3]  
## [1,]    2    0    0  
## [2,]    0    2    0  
## [3,]    0    0    2
```

```
i = diag(5,8,8);i
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,]    5    0    0    0    0    0    0    0  
## [2,]    0    5    0    0    0    0    0    0  
## [3,]    0    0    5    0    0    0    0    0  
## [4,]    0    0    0    5    0    0    0    0  
## [5,]    0    0    0    0    5    0    0    0  
## [6,]    0    0    0    0    0    5    0    0  
## [7,]    0    0    0    0    0    0    5    0  
## [8,]    0    0    0    0    0    0    0    5
```

```
# Experiment Below for 001, 010, 100
```

```
d = diag(1,2,2);d
```

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

```
d = diag(0,1,0);d
```

```
##  
## [1,]
```

```
d = c(1,0,0);d
```

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

```
d = c(0,1,0);d
```

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

```
d = c(0,0,1);d
```

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

```
# Experiment above
```

```
# Matrix Multiplication
```

```
m = matrix(c(2,6,5,1,10,4), nrow = 2, ncol = 3, byrow = TRUE);m
```

```
##      [,1] [,2] [,3]  
## [1,]    2    6    5  
## [2,]    1   10    4
```

```
t(m)
```

```
##      [,1] [,2]  
## [1,]    2    1  
## [2,]    6   10  
## [3,]    5    4
```

```
t = m %*% t(m);t
```

```
##      [,1] [,2]  
## [1,]   65   82  
## [2,]   82  117
```

```
# Create a 2*3 matrix A with values from 1 to 6 and a 3*2 matrix B with  
# values from 6 to 1  
# perform matrix mul of A and B  
# verify if result is a square matrix
```

```
m = matrix(1:6, 2,3);m
```

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

```
n = matrix(1:6, 3,2);n
```

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

```
v = m %*% n ;v
```

```
##      [,1] [,2]  
## [1,]   22   49  
## [2,]   28   64
```

```
# Verifying whether the output of matrix above is square matrix or not
nrow(v) == ncol(v)
```

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

```
# check if am is present in vector, also convert the same vector into matrix
```

```
v = c('hello','i','am',24,50,60,45,'rrr')
'am' %in% v
```

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

```
m = matrix(v, 2,4);m
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "hello" "am" "50" "45"
## [2,] "i"    "24" "60" "rrr"
```

```
# Create a 5*5 Matrix "u" with square of first 25 natural numbers
```

```
u = matrix(1:25, 5,5);u
```

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

```
u = u ** 2;u
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    36   121  256  441
## [2,]    4    49   144  289  484
## [3,]    9    64   169  324  529
## [4,]   16    81   196  361  576
## [5,]   25   100   225  400  625
```

```
#extract the sub matrix from 2:3 row and 2:3 columns
u[2:3,2:3]
```

```
##      [,1] [,2]
## [1,]   49  144
## [2,]   64  169
```

```
# Set diagonal elements to 0
diag(u) = 0;u
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0   36  121  256  441
## [2,]    4    0  144  289  484
## [3,]    9   64    0  324  529
## [4,]   16   81  196    0  576
## [5,]   25  100  225  400    0
```

```
#### DataFrame ####
```

```
# ?data.frame
```

```
v= c(14,26,38,30)
v_2 = c("Hello", 'Good Morning', "Nice to Meet you", "Me too")
d_1 = data.frame(v,v_2);d_1
```

```
##      v          v_2
## 1 14          Hello
## 2 26      Good Morning
## 3 38 Nice to Meet you
## 4 30          Me too
```

```
# Changing index names to user defined ones
```

```
D_1 = data.frame(v, v_2, row.names=c('A','B','C','D'));D_1
```

```
##      v          v_2
## A 14          Hello
## B 26      Good Morning
## C 38 Nice to Meet you
## D 30          Me too
```

```
# resetting index names back to 1 to 4
```

```
rownames(D_1) = 1:4
```

```
D_1; class(D_1)
```

```
##      v          v_2
## 1 14          Hello
## 2 26      Good Morning
## 3 38 Nice to Meet you
## 4 30          Me too
```

```
## [1] "data.frame"
```

```
rownames(D_1)[1] = 'ABC'; D_1
```

```
##      v          v_2
## ABC 14          Hello
## 2    26      Good Morning
## 3    38 Nice to Meet you
## 4    30          Me too
```

```
row.names(D_1)[1:2] = c('ABC', 'D');D_1
```

```
##      v          v_2
## ABC 14          Hello
## D   26      Good Morning
## 3   38 Nice to Meet you
## 4   30          Me too
```

```
row.names(D_1)[1] = 'E'
```

```
row.names(D_1)[c(1,4)] = c('X', 'Y'); D_1
```

```
##      v          v_2
## X 14          Hello
## D 26      Good Morning
## 3 38 Nice to Meet you
## Y 30          Me too
```

```
# ?head - To get top rows
```

```
head(D_1, 2)
```

```
##      v          v_2
## X 14          Hello
## D 26      Good Morning
```

```
head(D_1,-1)
```

```
##      v          v_2
## X 14          Hello
## D 26      Good Morning
## 3 38 Nice to Meet you
```

```
# ?View
```

```
# To have a look to table in larger format in dedicated window
```

```
View(D_1)
```

```
# ?tail - To get last rows
```

```
tail(D_1,2)
```

```
##      v          v_2
## 3 38 Nice to Meet you
## Y 30          Me too
```

```
# Structure - ?str - to get details of dim and datatype of that var
```

```
str(D_1)
```

```
## 'data.frame':   4 obs. of  2 variables:
## $ v : num  14 26 38 30
## $ v_2: chr  "Hello" "Good Morning" "Nice to Meet you" "Me too"
```

```
dim(D_1)
```

```
## [1] 4 2
```

```
# Generating Random Numbers  
sample(1:6,2) # Replace = True
```

```
## [1] 4 1
```

```
# 1st : end , numbers to be generated  
sample(1:6,7 ,replace = TRUE)
```

```
## [1] 2 1 6 5 1 6 2
```

```
set.seed(111)  
sample(1:5,1, replace = TRUE)
```

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

```
# Generating Random Names  
# install.packages("randomNames")  
# library(randomNames)  
# randomNames(20)  
  
## random selection from a vector  
  
stranger = c('Hemant', 'Onkar', 'Pratik', 'Bhupendra', 'Omkar',  
             'Bhupesh', 'Hitesh', 'Saurabh',  
             'Shubham', 'Narendra', 'Yash')  
friend = sample(stranger, 2);friend
```

```
## [1] "Bhupendra" "Pratik"
```

```
# Merging on the basis of ROWS
```

```
d = data.frame(XYZ = c(10,11,12,13,14), ABC = c('A', 'B', 'C', 'D', 'E'));d
```

```
##   XYZ ABC  
## 1  10  A  
## 2  11  B  
## 3  12  C  
## 4  13  D  
## 5  14  E
```

```
e = data.frame(XYZ = c(15,16), ABC = c('F', 'G'));d
```

```
##   XYZ ABC
## 1  10  A
## 2  11  B
## 3  12  C
## 4  13  D
## 5  14  E
```

```
d
```

```
##   XYZ ABC
## 1  10  A
## 2  11  B
## 3  12  C
## 4  13  D
## 5  14  E
```

```
rbind(d,e) # Binding on the basis of rows
```

```
##   XYZ ABC
## 1  10  A
## 2  11  B
## 3  12  C
## 4  13  D
## 5  14  E
## 6  15  F
## 7  16  G
```

```
str(d)
```

```
## 'data.frame':  5 obs. of  2 variables:
## $ XYZ: num  10 11 12 13 14
## $ ABC: chr  "A" "B" "C" "D" ...
```

```
# to find Row names in dataframe
names(d)
```

```
## [1] "XYZ" "ABC"
```

```
# to find column names in dataframe
colnames(d)
```

```
## [1] "XYZ" "ABC"
```

```
# To change datatype of column
d$ABC = as.factor(d$ABC)
str(d)
```

```
## 'data.frame':  5 obs. of  2 variables:
## $ XYZ: num  10 11 12 13 14
## $ ABC: Factor w/ 5 levels "A","B","C","D",...: 1 2 3 4 5
```

```
# To delete thw hole column
d$XYZ = NULL
d
```

```
##   ABC
## 1   A
## 2   B
## 3   C
## 4   D
## 5   E
```

```
# To change column name
colnames(d_1) = c("PQR", 'XYZ'); d_1
```

```
##   PQR          XYZ
## 1  14          Hello
## 2  26    Good Morning
## 3  38 Nice to Meet you
## 4  30          Me too
```

```
# change name of one column
colnames(d_1)[colnames(d_1) == 'PQR'] = 'XYZ'; d_1
```

```
##   XYZ          XYZ
## 1  14          Hello
## 2  26    Good Morning
## 3  38 Nice to Meet you
## 4  30          Me too
```

```
colnames(d_1)[1] = 'UVW';d_1
```

```
##   UVW          XYZ
## 1  14          Hello
## 2  26    Good Morning
## 3  38 Nice to Meet you
## 4  30          Me too
```

```
# add ing new column to dataframe
```

```
d_1$newcolumn = c('s','t','u','v'); d_1
```

```
##   UVW          XYZ newcolumn
## 1  14          Hello        s
## 2  26    Good Morning        t
## 3  38 Nice to Meet you        u
## 4  30          Me too        v
```

```
# Create a DataFram "df" with columns "ID", "Name", "Age"
# Add 20 enteries
# Rename the "Name" to column to "FullName"
```



```

# Drop the ID Column from the DataFrame
# Display the names of the remaining column

library(randomNames)

df = data.frame(ID=c(1:5), Name =randomNames(20), Age=sample(21:30, 5));df
print("1")
colnames(df)[2] = 'FullName' ; df

print("2")
df$ID = NULL;df

print("3")
colnames(df)

# Create a DataFrame "Sales" with column "Product"(5 unique names) and "Price"
# (random values between 10 & 50)
# Add a new column "Discount" that is 10% of the price
# Create another column "FinalPrice" by sustracting
# "Discount" from "Price"

Sales = data.frame(Product = c("BMW", 'Mercedes', 'Maserati',
                                'Ferrari', 'Porsche'),
                    Price = sample(10:50, 5));

Sales$Discount = Sales$Price*0.10;

Sales$FinalPrice = Sales$Price - Sales$Discount ; Sales

# Create a dataframe "EMployee" with "Name", "Age", & "Salary"
# columns for 10 EMployees
# filter rows where age is above 30
# select only the name and salary columns
# Count the number of employees with salary greater than 50,000

Employee = data.frame(Name = randomNames(10), Age = sample(20:40,10),
                      Salary = sample(10000:75000, 10)); Employee

which(Employee$Age > 30); Employee

ageabove = Employee[Employee$Age>30, ]; ageabove

name_salary = Employee[,c("Name", "Salary")]; name_salary

salary_more_50 = sum(Employee$Salary > 50000); salary_more_50
Employee[Employee$Salary>50000, ]

```