

Practical – 05

Title: - Introduction to R programming and Data acquisition.

Aim: - To understand R Graphics and how to import data.

Print Operation

```
print("Hello World") [1] "Hello World" 5+5+0+98 [1] 108 32+87 [1] 119
```

Add values to variables

```
name <- "Onkar Malawade" name [1] "Onkar Malawade"
var1 = 1234567 var1 [1] 1234567
var2 <- 987654 var2 987654
res <- var1 + var2 res 2222221
res <- var1 res 1234567
res <- var1 * var2 + var1 res 1.219326e+12
res = res / 100000098 res 12193.25
```

String Concatenation

```
text1 = "Hello!!!" text2 = "Onkar!!!" paste(text1,text2) "Hello!!!
Onkar!!!" paste(text1,text2,"Malawade") "Hello!!! Onkar!!! Malawade"
```

For Loop

```
for (x in 1 : 10){ print(x) }
[1] 1 [1] 2 [1] 3 [1] 4 [1] 5 [1] 6 [1] 7 [1] 8 [1] 9 [1] 10
```

Class or we can say Datatypes of given variables from the class()

method:-

```
x<-10.5 class(x) [1] "numeric"
x<-1000L class(x) [1] "integer"
x <- 9i + 3 class(x) [1] "complex"
x <- "R is Exists" class(x) [1] "character"
x <- TRUE class(x) [1] "logical"
```

Mathematical functions

```
max(10,23,32) [1] 32
min(12,1,3) [1] 1
abs(-4.12) [1] 4.12
sqrt(25) [1] 5
ceiling(4.5) [1] 5
floor(4.5) [1] 4
str = "HelloWorld"; nchar(str) [1] 10
```

Checking a String

```
str = "HelloWorld"; nchar(str) [1] 10
grepl("H",str) [1] TRUE
grepl("Hello",str) [1] TRUE
grepl("Onkar",str) [1] FALSE
```

Formatting

```
res1 = format("Onkar",width=8,justify="l") res2 =  
format("Onkar",width=8,justify="c") res3 =  
format("Onkar",width=8,justify="r")  
print(res1) [1] "Onkar "  
print(res2) [1] " Onkar "  
print(res3) [1] " Onkar"
```

Use of nsmall Format

```
format(13.7) [1] "13.7"  
format(13.7,nsmall=3) [1] "13.700"  
format(13.7,nsmall=2) [1] "13.70"  
format(13.7,6.07,digits=2) [1] "14"  
format(13.7,6.07,digits=2,nsmall=1) [1] "13.7"
```

Use of Scientific Format

```
format(2^31-1) [1] "2147483647"  
format(2^31-1,scientific=TRUE) [1] "2.147484e+09"
```

Use of Text Formatting

```
text1 = "Onkar"  
toupper(text1) [1] "ONKAR"  
tolower(text1) [1] "onkar"
```

Creating a Function

```
myFunction <- function(){ //create a function with the name myFunction  
print("Hello World!!!")  
}
```

Call Function

```
myFunction()
```

Output: Hello World!!!

Arugments

The arguments gives information to the function

Default Parameters

```
> myFunction <- function(country = "Pune"){  
+ paste("I am from ",country)  
+ }  
> myFunction()  
[1] "I am from Pune"  
> myFunction("Talere")  
[1] "I am from Talere"  
> myFunction("Bhandarpule")  
[1] "I am from Bhandarpule"  
>
```

With Parameters

```
myFunction <- function(fname){  
+ paste(fname,"Onkar")  
+ }  
myFunction()
```

Error in myFunction() : argument "fname" is missing, with no default

```
myFunction("Atharav")  
[1] "Atharav Onkar"  
myFunction("I am ")  
[1] "I am Onkar"
```

Functions

```
> add <- function(num1 , num2){  
+ return (num1 + num2)  
+ }  
> print(add(10, 20))  
[1] 30  
> add(10, 20)  
[1] 30  
> mult <- function(num1){  
+ return (num1 * 10)  
+ }  
> mult(10)  
[1] 100  
>
```

Vectors

```
> fruits <- c("banana","apple","orange")  
> fruits  
[1] "banana" "apple" "orange"  
>
```

Sequence

```
> print(seq(5,9,by=0.4))  
[1] 5.0 5.4 5.8 6.2 6.6 7.0 7.4 7.8 8.2 8.6 9.0  
> num = seq(from = 0, to = 100, by = 20)  
> num  
[1] 0 20 40 60 80 100  
>
```

Using c() function : Vector with String Values

```
> s <- c('apple','red',5,TRUE)  
> s  
[1] "apple" "red" "5" "TRUE"  
>
```

Using (:) = Vector with Numeric Values

```
> numbers <- 1:10
> numbers
[1] 1 2 3 4 5 6 7 8 9 10
> numbers <- 1.4 : 5.4
> numbers
[1] 1.4 2.4 3.4 4.4 5.4
> numbers <- 1.4 : 5.6
> numbers
[1] 1.4 2.4 3.4 4.4 5.4
>
```

Vector Length() Fuction

```
> fruits <- c("banana","apple","orange")
> length(fruits)
[1] 3
```

Sort a Vector using sort() function

```
> fruits <- c("banana","apple","orange","lemon","grapes")
> sort(fruits)
[1] "apple" "banana" "grapes" "lemon" "orange"
> numbers <- 1.4 : 5.6
> sort(numbers)
[1] 1.4 2.4 3.4 4.4 5.4
>
```

Access Vectors Using [index] brackets

```
> fruits <- c("banana","apple","orange","lemon","grapes")
> fruits[1]
[1] "banana"
```

Access Vectors Using [From_index,ToIndex] brackets

```
> fruits <- c("banana","apple","orange","lemon","grapes")
> fruits[c(1,4)]
[1] "banana" "lemon"
>
```

Change an item in Vector

```
> fruits[2] <- "Peanuts"
> fruits
[1] "banana" "Peanuts" "orange" "lemon" "grapes"
>
```

Repeat Vectors each value

```
> repeat_each <- rep(c(1,2,3),each=3)
> repeact_each
```

Error: object 'repeact_each' not found

```
> repeat_each
[1] 1 1 1 2 2 2 3 3 3
```

Repeat sequence of the vector

```
> repeat_times <- rep(c(1,2,3),times=3)
> repeat_times
[1] 1 2 3 1 2 3 1 2 3
>
```

Repeat Each value Independent

```
> repeat_indep <- rep(c(1,2,3),times=c(5,3,1))
> repeat_indep
[1] 1 1 1 1 1 2 2 2 3
>
```

Vectors with Arithmetic Operations

```
> v1 <- c(2,4,6,8,7)
> v2 <- c(5,4,6,3,7)
> res = v1 + v2
> res
[1] 7 8 12 11 14
```

```
> res = v1 - v2
> res
[1] -3 0 0 5 0
```

```
> res = v1 * v2
> res
[1] 10 16 36 24 49
```

```
> res = v1 / v2
> res
[1] 0.400000 1.000000 1.000000 2.666667 1.000000
```

R Factors :

Factors are the data objects used to categories data and store it as levels.

They can store both strings and integers.

To create a factor, use factor() function and add a vector as argument:

```
> music <-
factor(c("Jazz","Rock","Pop","Classic","Classic","Pop","Rock","Jazz","Jazz"))
> music
[1] Jazz Rock Pop Classic Classic Pop Rock Jazz Jazz
Levels: Classic Jazz Pop Rock
Use levels() function
> levels(music)
[1] "Classic" "Jazz" "Pop" "Rock"
> levels = c("Jazz","Pop","Classic","Rocks","Others")
> levels(music)
[1] "Classic" "Jazz" "Pop" "Rock"
```

Use Factor length() Function

```
> length(music)
```

```
[1] 9
```

```
>
```

Access Factors

```
> music[2]
```

```
[1] Rock
```

```
Levels: Classic Jazz Pop Rock
```

```
>
```

R Data Frames :

To Create Data Frames in data.frame() function.

It is Table like structure.

```
> Data_frame <- data.frame(Taining =  
c("Strength","Stamina","Others"),Pulse=c(100,150,120),Duration=c(60,3  
0,45))
```

```
> Data_frame
```

```
Taining Pulse Duration
```

```
1 Strength 100 60
```

```
2 Stamina 150 30
```

```
3 Others 120 45
```

Summary of the Data

```
> summary(Data_frame)
```

```
Taining Pulse Duration
```

```
Length:3 Min. :100.0 Min. :30.0
```

```
Class :character 1st Qu.:110.0 1st Qu.:37.5
```

```
Mode :character Median :120.0 Median :45.0
```

```
Mean :123.3 Mean :45.0
```

```
3rd Qu.:135.0 3rd Qu.:52.5
```

```
Max. :150.0 Max. :60.0
```

```
>
```

Access Data Frames

```
Data_frame[1]
```

```
Training 1 Strength 2 Stamina 3 Others
```

```
Data_frame[["Pulse"]]
```

```
Data_frame$Duration
```

Add Rows

```
New_Row_DF <- rbind(Data_frame,c("Strength",110,102))
```

```
New_Row_DF Taining Pulse Duration 1 Strength 100 60 2 Stamina 150  
30 3 Others 120 45 4 Strength 110 102
```

Add Columns

```
New_col_DF <- cbind(Data_frame,steps=c(121,112,122))
```

```
New_col_DF
```

```
Taining Pulse Duration Steps 1 Strength 100 60 121 2 Stamina 150 30  
121 3 Others 120 45 112 >
```

Remove Rows and Columns

```
Data_new_frame <- Data_frame[-c(1),-c(1)]
```

```
Data_new_frame
```

```
Pulse Duration 2 150 30 3 120 45
```

Amount of Rows and Columns

```
dim(Data_frame)
```

```
3 3
```

```
ncol(Data_frame)
```

```
3
```

```
nrow()
```

```
3
```

Get Structure of Data Frame

```
str(Data_frame)
```

```
'data.frame': 3 obs. of 3 variables: $ Taining : chr "Strength" "Stamina"  
"Others" $ Pulse : num 100 150 120 $ Duration: num 60 30 45
```

R - import Data:

```
data <- read.csv("input.csv")
```

```
data
```

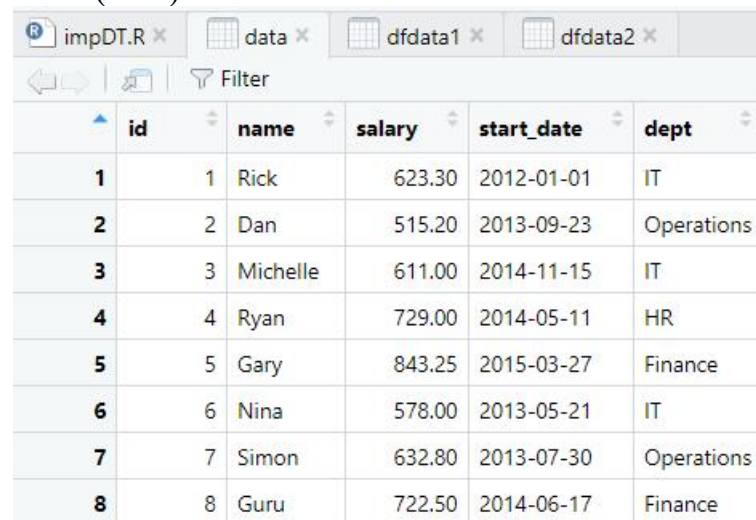
```
data$dept
```

```
[1] "IT"      "Operations" "IT"      "HR"      "Finance" "IT"      "Operations"  
[8] "Finance"
```

```
data <- read.csv("data.csv")
```

```
data
```

View(data)



	id	name	salary	start_date	dept
1	1	Rick	623.30	2012-01-01	IT
2	2	Dan	515.20	2013-09-23	Operations
3	3	Michelle	611.00	2014-11-15	IT
4	4	Ryan	729.00	2014-05-11	HR
5	5	Gary	843.25	2015-03-27	Finance
6	6	Nina	578.00	2013-05-21	IT
7	7	Simon	632.80	2013-07-30	Operations
8	8	Guru	722.50	2014-06-17	Finance

```
nrow(data)
```

```
[1] 8
```

```
> dim(data)
```

```
[1] 8 5
```

```
[2] >
```

```
> names(data)
```

```
[1] "id"      "name"    "salary"  "start_date" "dept"    >
```

```
> rownames(data)
```

```
[1] "1" "2" "3" "4" "5" "6" "7" "8"
```

```
dfdata = select(data,'Country','Age','Purchased')
dfdata
```

```
Country Age Purchased
1 France 44 No
2 Spain 27 Yes
3 Germany 30 No
4 Spain 38 No
5 Germany 40 Yes
6 France 35 Yes
7 Spain NA No
8 France 48 Yes
9 Germany 50 No
10 France 37 Yes
```

```
dfdata1 = filter(dfdata,Country=='France')
View(dfdata1)
```

	Country	Age	Purchased
1	France	44	No
2	France	35	Yes
3	France	48	Yes
4	France	37	Yes

```
dfdata2 = filter(dfdata,Country=='France',Age<=40)
View(dfdata2)
```

	Country	Age	Purchased
1	France	35	Yes
2	France	37	Yes

```
is.na(NA)
TRUE
```

```
> sum(is.na(data))[1] 2> > apply(data,is.numeric) No Country Age Salar
y Purchased
```

```
TRUE FALSE TRUE TRUE FALSE >> sum(data$Age,na.rm = TRUE)
E)[1] 349>> View(data)>> data$Age <- ifelse(is.na(data$Age),ave(data$Age,FUN
= function(x) mean(x,na.rm=TRUE)),data$Age)>> View(data)>> data No Countr
y Age Salary Purchased
```

```
1 1 France 44.00000 72000 No
2 2 Spain 27.00000 48000 Yes
3 3 Germany 30.00000 54000 No
4 4 Spain 38.00000 61000 No
5 5 Germany 40.00000 NA Yes
6 6 France 35.00000 58000 Yes
7 7 Spain 38.77778 52000 No
8 8 France 48.00000 79000 Yes
9 9 Germany 50.00000 83000 No
10 10 France 37.00000 67000 Yes
```

```
> data$Salary <- ifelse(is.na(data$Salary),ave(data$Salary,FUN = function(x) mean
```



```
(x,na.rm=TRUE)),data$Salary)
```

```
>
```

```
> View(data)
```

```
>
```

```
> data
```

	No	Country	Age	Salary	Purchased
1	1	France	44.00000	72000.00	No
2	2	Spain	27.00000	48000.00	Yes
3	3	Germany	30.00000	54000.00	No
4	4	Spain	38.00000	61000.00	No
5	5	Germany	40.00000	63777.78	Yes
6	6	France	35.00000	58000.00	Yes
7	7	Spain	38.77778	52000.00	No
8	8	France	48.00000	79000.00	Yes
9	9	Germany	50.00000	83000.00	No
10	10	France	37.00000	67000.00	Yes