

## Data Science For Engineers

### Week - 1

In R Studio,

- Console panel (towards left) - type commands & see results
- Environment / History pane (top right) - variables generated in temporary workspace & all the commands used till now
- Files / Plots / Packages / Help / Viewer (bottom right)

Working directory → can be set by 2 methods

- (i) Using console - command setwd("path") is used
- (ii) Using GUI

# We need to save the file before we can run the program.

→ Source command executes all commands in console without displaying them.

WHEREAS

→ Source with echo will execute & print automatically

→ Run command can be used to execute selected lines.

## Adding comments -

- (i) Single ~~st~~ line comment → #
- (ii) Multiple lines

↳ Use GUI  
↓

Select lines to be commented  
Then press `ctrl + shift + c`

Clearing the console → `ctrl + L`

Clearing the environment →

- (i) single variable → `rm(variable)`
- (ii) All variables → `rm(list = ls())`

## Saving -

- `save(a, file = "sess1.Rdata")` - to save a single variable `a`
- `save(list = ls(all.names = TRUE), file = "sess1.Rdata")` - to save a full workspace
- `save.image()` - shortcut function to save whole workspace
- `load(file = "sess1.Rdata")` - to load saved workspace
- `"typeof()"` → find data type of object
- prefix `"is."` → verify if object is of certain datatype
- prefix `"as."` → convert data type of object to another.

Vector  $\rightarrow$  ordered collection of same data type

List  $\rightarrow$  ordered collection of objects

Data frame  $\rightarrow$  generic tabular object

All the elements ~~was~~ of a vector must be of same data type

"c" is used to concatenate

Accessing components  $\rightarrow$

To access top level components, use double slicing operator "[[ ]]" or "[ ]" and for lower/inner level components, use "[ ]" along with "[[ ]]".

Eg: `print(emp.list[2])`

Output: "Man" "Raj" "Sha" "Din"

Eg: `print(emp.list[[2]][1])`

Output: "Man"

Concatenation of lists  $\rightarrow$

Concatenation function "`c(list1, list2)`" can be used to concatenate two lists.

Dataframes → are generic data objects of R  
used to store tabular data

```
df = data.frame(vec1, vec2, vec3)  
print(df)
```

```
newDF = read.table(path = "path of file")
```

Accessing rows & columns →

Accessing first & second row →  

```
print(df[1:2,])
```

Accessing first & second column →  

```
print(df[, 1:2])
```

ALTERNATE → 

```
print(df[1:2])
```

Subset → `subset()` extracts subsets of data  
based on conditions.

"|" → or

Eg: a|b → a or b

- Dataframes can be edited by direct assignment
- It can also be edited using `edit()` command.

Adding extra rows & columns →

Extra row → "rbind" function

Extra column → "cbind" function

Deleting rows & columns →

A '-' sign before value & before ',' for rows and after ',' for columns

"|" means no to those rows/columns which satisfy the condition.

Recasting dataframes →

Recasting is the process of manipulating a data frame in terms of its variables

Two steps → (i) Melt

(ii) Cast

Applying the recast() function performs melt & cast in one command.

Adding new variable to dataframe based on existing ones →

"mutate()" command will do so

Combining two dataframes →

"function(dataframe1, dataframe2, by = id.variable)"

- ✓ left-join()
- ✓ right-join()
- ✓ inner-join()
- fulljoin()
- semi-join()
- anti-join()

Arithmetic operations in R →

Symbol	Operation
=, <-	Assignment
+	Addition
-	Subtraction
*	Multiplication
/	Division
^, **	Exponent
% %	Remainder
%/%	Integer Division

→ In R, only '<-' is valid, but in R Studio, '=' works too.

Hierarchy of operations →

Bracket	()
Exponent	^, **
Division	/
Multiplication	*
Addition & Subtraction	+, -

Creating Matrices →

$A = \text{matrix}()$

$A = \text{matrix}(c(1, 2, 3, 4, 5, 6, 7, 8, 9), \text{nrow} = 3, \text{ncol} = 3, \text{byrow} = \text{TRUE})$

byRow is FALSE by default

Dimension / Size →  $\text{dim}(A)$

No. of rows →  $\text{nrow}(A)$

No. of columns →  $\text{ncol}(A)$

Number of elements →  $\text{length}(A)$

where  $A$  is a matrix

Row →  
→ ↓ ↓ Column

$1:10 \rightarrow 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

$10:5 \rightarrow 10, 9, 8, 7, 6, 5$

$A * B \rightarrow$  Element wise multiplication

$A \%*\% B \rightarrow$  Regular matrix multiplication

$A / B \rightarrow$  Element wise division

Functions in R take multiple input objects but returns only one object as output.

Looping over objects →

Looping functions →

- (i) `apply` : Apply a functions over the margins of an array or matrix
- (ii) `lapply` : Apply a function over a list or a ~~matrix~~ vector
- (iii) `tapply` : Apply a function over a ragged array.
- (iv) `mapply` : Multivariate version of `lapply`
- (v) `xxply` : (plyr package)

Scatter Plot →

```
X = 1:10
```

```
Y = X^2
```

```
plot(Y)
```

Line plot →

```
X = 1:10
```

```
Y = X^2
```

```
plot(X, Y, type = 'l')
```



Bar plot →

`barplot(H, names.arg, nlab, ylab, main, names.arg, col)`

Eg: `barplot(H, names.arg = M, nlab = "Month", ylab = "Revenue", col = "blue", main = "Revenue chart", border = red)`