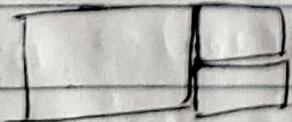


① Introduction to R studio :

- R studio is available as both open source & commercial software.
- R studio is also available on various platforms like windows , Macos , Linux
- R studio is used for desktop as well as server versions.
- R studio is the open source that provide IDE for R language.
- R studio can be downloaded from official website <https://rstudio.com/>



- Left panel is where R is waiting for you to tell it what to do. & see the result generated.

- Top right top corner is the history panel
- Bottom " another panel which contains multiple tabs such as files, packages, environment, triggers, (but using RStudio)

* Basis of R programming :

- vectors
- loops
- functions
- probability

* Strings in R programming :-

strings must start / end with same quotes and

Eg: valid string "hello", "hi", "he's"

Invalid string 'hello', "bye" he"

e ← 'mix'

print(e) ⇒ mix is printed

b ← "hi"

print(b) ⇒ ngt is printed.

* Variable can be assigned value in 3 ways:

var-name ← Value # leftward operator

" = Value # assignment

" → Value # rightward operator

→ To see the data type of a variable we use class()

function

→ to see list of variables we use ls() func.

→ " delete a variable we use rm() func.

Eg: var ← 15

class(var) # prints 15

vary ← "hello" #

class(vary) # prints hello.

Vectors :

- It is basic R data object
- characters "ABC" Raw → `charToRaw("val")`
- complex 2+8i • Integer 45L
- Double 12.5 • Logical T/F

Add of 2 vectors

```
v1 <- c(3, 8, 4, 8, 9, 11)  
v2 <- c(4, 11, 16, 5, 2, 1)  
add.result <- v1 + v2  
print(add.result).
```

Factors :

- are used in lots of repeating values.
- factors are created using `factor()` function.

Syntax : `factor(data)`

Eg :

```
fac.data <- c ("East", "W", "N", "S", "E", "W")  
fac.data <- factor(fac.data).  
print(fac.data).  
#> [1] East, W, N, S.
```

: concatenation of lists :

`c(list1, list2)`

Eg : `emp.ages = list("ages" = c(23, 48, 54, 30, 32))`

`emp.list = c(emp.list, emp.ages)`

`print(emp.list)`



Vector operation:

```
rec = c(0, 3, 8, 1, 6)
```

```
print(rec)
```

```
rec1 = c(8, 5, 1, 11, 10, 2, 5)
```

```
print(rec1)
```

```
v = rec + rec1
```

```
print(v)
```

Now the length of rec < rec1 (so rec will be recycled).

Same goes with + -.

Read the CSV file:

```
→ input_data <- read.csv("C:/Users/1ddeboll/Desktop/abc.csv", header = FALSE)
```

```
print(input_data)
```

header = FALSE means column is treated as data

Excel:

- In the Environment / history Section there will be import from excel.

- With the help of excel we can write code.

```
library(readxl) is a package
my_data <- read_excel("Dataフレンズ.xlsx",
                      sheet = "position", range = "C4:G14", na = "***")
```

Inbuilt datasets:

- data()

- co2

- names(mtcars) → If we want

- mtcars

- tree

- to extract name

- iris

- USArrests

- head(mtcars) → headings
will be printed.

- nrow(mtcars) → no. of rows
will be printed.

• Reading CSV file

• Reading a txt file : This can be anything
chlorine <- read.table(file.choose(), header = TRUE,
sep = "\t") # sep for tab

head(chlorine) # displays 1st 6 rows.

Another method to display txt file

chlorine <- read.table("~/Desktop/chlorine.txt", header =
"TRUE", sep = "\t")

• Writing a txt file :

Date format in R

my.date <- as.Date("2020-05-07")

my.date # prints my.date data in seq. format

format(my.date, "%d-%m-%y")

"[my.date, "%y-%m-%d")] in [month] diff format

(my.date, "%d-%m-%y-%H-%M-%S")

"[my.date, "%Y/%m/%d")] in [year] format

• Regular Expressions in R.

→ Sequence of characters used to search text.

Eg :

→ Searching file in a directory using command line

→ Replacing Specific text.

RE Examples

Pattern	Text Matches	Reg	Text	Explanation
w+	Any alphanumeric	grexpl(pattern='[w\Khtn]')	a	
d+	" " digit		1	
w+	word to apply sub		(un) word	
d+			1234	
s	Space			
S	Any non-space		word	

* Packages :

- R || are a collection of sample data & functions
- while download R studio some default packages will be installed.

i.e library(): gives list of all available packages

install.packages(): To install new package.

library("package Name", lib.loc = "path of lib"): to load package to lib.

• libPaths() → give path of libraries

• to check currently loaded lib we use search()

• install.packages("XML")

* External packages :

Syntax : install.packages() → list of packages will be displayed, we can choose appropriate one & execute. List of package might increase bcz R is a open source



Q) CRAN :

- centralized Radio Access Network
- In effort to create cloud computing ~~atmos~~ architecture for support of 5G Network.
- It is made of Base Band Unit (BBU), Remote Radio Unit (RRU) & transport network that is also called fronthaul.
- BBU functions as a cloud / data center.
- RRU connects wireless devices.

Q) Downloading & Installing packages from CRAN :

- on R's main package repository CRAN alone have over 10,000 packages available to choose from
- To install " " from CRAN we need to type `install.packages()`
- The type of package we need to install should be mentioned inside common braces.
Eg: `dplyr`.
then : `install.packages("dplyr")`
- This will install dplyr along with all its dependencies.
- If we execute the above line outside RStudio.
Eg: R's built in GUI / Terminal we need to choose CRAN mirror
- The common of all that can be selected is (o-cloud).