R

R lang - very similar ~~to~~ appearance to "s"-langeuage based on a software S-plus

R foundation → non-profit orgn in the public interest

Professors - Ross Ihaka
Robert Gentleman } university of Auckland, New Zedand

in 1991 beagan impn of S lang without S software

- 1st official announce (release) - 1991
- Comprehensive R Archive Network (CRAN) - 23 April 1997 with 8 mirrors & 12 contributed pkgs

→ 1st stable beta V1.0 → 29 Feb 2000

→ Nov, 2020 >16000 pkgs -available.

→ R

1. free (open source) s/w
2. convinient to use for statn and graphical applications
3. available for windows, Linux, Unix, Mac pfs
4. Interpreted language
5. useful for computation, simulation and managing dbs

Getting help in R

→ to get help with a function → ?read.table

find(a) → function returns a package that a is in.

apropos("lm") - returns a character vector giving the names of all objects in search list that match your enquiry

op: [1] "colMeans"

Demonstration of R functions

demo(persp) demo(graphics)

↑ cmd for 3 surface plots

→ example("lm") provides help on finding an example on the function lm

Week-2 1st vd skipped

Lecture-6

ls() → get contents of my comput

getwd() → finds current working directory in R

setwd() → to change working directory in R

Esc → to interrupt a long-running computation and return to the command prompt without exiting.

Ctrl+ L → to clean the GUI window in R

RSiteSearch("keyword") - to search the web for information and answers Regarding R

When u down the computer and R is restarted, we need to reload the libraries again

5.2345e+7 = 5.2345x10^7

rm() → command removes variables names

detach() → detaches objects from the Search Path

It removes from search() path of available R objects

To get rid of everything, include data frames, type rm(list = ls())

q() → to quit R

lecture-7

Assignment operators ← (or) =

characters — within ' ' or " "

is.numeric() returns TRUE or FALSE

is. character()

→ as.numeric() → to convert value as a number

as. character() → to convert value as a character

x=20

y= as.charcter(x)

y

x = "apple"

z = is.numeric(x)

op: Warning

NAs intiod by coercion

comments #

R→case sensitive

Data vector combines various numbers into a vector

```
> x = c(1,2,3,4,5)
> x
op: 1 2 3 4 5
```

mode() → explains type or storage mode of an object

modes available in R

"logical"
"integer"
"double"
"complex"

"raw"
"character"
"list"
"expression"

"name"
"symbol"
"function"

storage.mode()	mode() ~~sto~~
integer	numeric
double	numeric

is.finite()
is.infinite()

16 17 18 19
20 21 22 23
24 25 26 27
28 29 30 31

Lecture 8 R as a calculator

addition of data vector and scalar

```
> c (2,3,5,7) + 10
op:  12  13  15  17
```

10 40 70
20 50 80
30 60 90

1 2
3 4

5 6
7 8

lecture-9

Libraries in R

if L not included in base package, it can be downloaded and loaded
↑
library

To use a library → library(spatial)

Ex of libraries that comes as a part of base package to R : 1. MASS (Modern Applied Statistics using S-Plus)
2. mgcv

that do not come
→ 1. spatial
2. boot

packageDescription() → provides description file of a package
packageDescription("spatial")

help → library(help = spateal)

To install
installopackages("boot")

To check which packages are installed
installedo packages ()

To uninstall , removeo packages("packages")
To update updateopackages ("cluster")
To unload packages , detach("package: cluster", unload=TRUE)

Lecture 9

Addition : ① c(12,3,5,7) + c(-12,-3,-5,8)

= c(0,0,0,15)

② c(12,3,5,7) + c(-12, -3)

= 12 3 5 7
-12 -3 -12 -3 ← repetition of 2nd vector

= c(0,0,-7,4)

* ③ c(2,3,5,7) + c(1,2,3)

= c(3, 5, 8,8) and warning.

length of longer vector ≠ shorter vector length multiple

Subtraction ① c(12,3,5,7) - c(12,3,5,6)

= c(0,0,0,1)

② c(12,3,5,7) - c(1,2) = c(11,1,4,5)

Multiplication ① c(2,3,5,7) * c(-2,-3,-5,8)

= c(-4,-9,-25,56)

② c(2,3,5,7) * c(8,9,10) = 16 27 50 56

Division also same

Lecture 10

Power operations

2^3 ⇒ 8

2**3 ⇒ 8

$\frac{1}{\sqrt{2}}$ = 2^-0.5 ⇒ 0.707

Integer division with scalar

2 %/% 2 = 1

5 %/% 2 = 2

c(2,3,5,7) %/% 2 = 1 1 2 3

c(2,3,5,7) %/% c(2,3) = 1 1 2 2 (Int div with data vector)

modulo division with scalar

2 %% 2 = 0

3 %% 2 = 1

7 %% 3 = 1

with data vectors

c(2,3,5) %% c(2,3)

> 0 0 1

Warning

Lecture-11

Builtin functions

→ max(1,2,3)

> 3

max(c(1,2,3))

> 3

→ min()

→ mean()

→ abs()

→ sqrt()

→ round()

→ ceiling()

→ floor()

→ sum() , prod()

→ log() , → log10() , log2()

→ exp()

→ sin() , cos() , tan()

→ sinh() , cosh()

Lecture - 12

Matrices

rectangular array with p rows & n col

x = matrix(nrow=4, ncol=2, data = c(1,2,3,4,5,6,7,8))

default byrow=F)

```
> x
        [,1]   [,2]
[1,]     1      5
[2,]     2      6
[3,]     3      7
[4,]     4      8
```

dim(x)

⇒ returns nrows ncol

nrow(x)

⇒ returns no of rows

ncol(x)

⇒ returns no of cols

mode(x)

⇒ informs type/ storage mode of obj numerical/logical

help("matrix)

Sums

c(4,4,1,0.6667)

c(12,16,3,2) / c(3,4,3,3) +

c(4,4,64) - c(4,7,6,8) +

c(9, -8, -6, -4)

6, 6, 7, 4 + c(144, 168, 144, 168) +

sqrt(c(12,6,7,8) * c(3,6,7,2) → error

c(3, 4, 4, 4) + c(6, 8, 5, 5) + c(7,3,4,0)

16 15
13 9

Lecture - 13, Matrix operations

rownames by default → [1,] [2,] [3,] . .

to rename → rownames(matrixname) = c("r1","r2","r3")

to rename columnname

colnames(x) = c(" c1", "c2", "c3")

Assigning specified no. to all matrix elements

x = matrix(nrow=4, ncol=2, data=2) ⇒

	[,1]	[,2]
[1,]	2	2
[2,]	2	2
[3,]	2	2
[4,]	2	2

diagonal matrix

	[,1]	[,2]	[,3]
[1,]	1	0	0
[2,]	0	1	0
[3,]	0	0	1

d = diag (1, nrow=3, ncol=3)

.. (c(1,2,3) " ..

>
1 0 0
0 2 0
0 0 3

lec-14

Transpose of a matrix x = x'

command → t

→ xt = t(x)

ex: x→ 1 5
2 6
3 7
4 8

→ rowSums(x) → finds sum of no. in rows ✓ > 6 8 10 12

→ colSums(x) > 10 26

→ rowMeans(x) > 3 4 5 6

→ colMeans(x) > 2.5 6.5

x ←

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	4	5	6
[3,]	7	8	9
[4,]	10	11	12
[5,]	13	14	15

x[c(1,4) , c(1,3)]

>

	[,1]	[,2]
[1,]	1	3
[2,]	10	12

x+5 → 5 is added to each element in the matrix

x*5 → multiplied

Lecture-15

x %*% y → multiplication of 2 matrices
4x2 2x4

crossproduct = x' x can be obtained using crossprod(x)

concatenating matrices rowwise : rbind(x,y)
colwise : cbind(x,y)

inverse of a matrix → solve(x)

eigen() finds eigen values and eigen vectors of matrix

lec-16 logical operators >, >=, <, ==, !=, !, &&, &, |, ||
AND

xor(x,y) isTRUE(x) , isFALSE (x)

&, | → performs element-wise comparisons in almost same way as arithmetic operators

&&, || → evaluates left to right examining only the 1st element of each vector. Evaluation proceeds until result is determined

x = c(8,18)

(x<10) || (x<2)

TRUE

operates only on first element and not on remaining elements

(x<10) | (x<2)

TRUE FALSE

operates on all elements

Sum

x = 1:6

x > 2 & (x<5) => F F T T F F

x[(x > 2) & (x<5)] => 3 4

Lecture-17

is.logical(x) → checks whether a variable is logical or not

x = c(1,2,3)

y = c(4,5,6)

x > y # compares 1>4 , 2>5 , 3>6 => FALSE FALSE F

x < y => TRUE TRUE TRUE

x!=y => TRUE TRUE TRUE

isTRUE(8<6) => F   isFALSE(5>8) => T

Lecture-18 Missing data handling

is.na() → TRUE if ele loc contain missing values rep by NA

① x = NA
is.na(x)
TRUE

② x = c(11, NA, 13, NA)
is.na(x)
FALSE TRUE FALSE TRUE

③ x = c(11, NA, 13, NA)
mean(x)
> NA

④ mean(x, na.rm = TRUE)
> 12

NA → placeholder for something that exists but is missing

NULL → stands for something that never existed at all.

x = c(11,NA,13,NA)

which(is.na(x)) → location of missing values

2 4

sum(is.na(x)) → returns count of NAs

complete.case(x) → represents which are not missing

> TRUE FALSE TRUE FALSE

y=na.omit(X) → returns obj with listwise deletion of missing values

```
> y
  11 13
 attr( , "na.action")
   2 4
 attr( , "class")
   "omit"
```

Lecture-19,20,21 Conditional executions - If and If-Else

```
if (x>5)  x = x-1  else  2*x
if (x>5) {x = x-1}  else  {2*x}

ifelse( test, yes, no)

switch( 1, "apple" ,"banana"  , "orange")
  > apple
```

which( ) ← works like search operator

```
x = c(10,15,8, 14,6,12)
> which (x==14)       //  4
> which(x!=12)        //  1 2 3 4 5

x =   1   4   7            > which.min(x)   // 1
      2   5   8
      3   6   9            > which (x%%2==1)   // 1 3 5 7 9

                          > which( x%%2==1)  // returns pos
                                                 (r,c) of
                 row  col                          value
          [1,]    1    1
            2     3    1
            3     2    2
            4     1    3
            5     3    3
```

Lec 22,23 loops

for (name in vector) ex: for(i in 1:5) { print(i)}
while (condi) { }
repeat

paste(x, y) //returns "Hello_World"
"Hello" "World"

'next' in R = continue in python

Lecture-24 Function - bunch of cmds grouped together in a senctble unit

→ components of a function : function name , args , body , return va

→ built-in functions : sum() , prod() , mean() , max(), sum(x)

Syntax

```
name <- function(arg1, arg2, ..) {
 (or)
  =        expression(s)
          }
```

lecture-25 Sequences : set of related numbes , events, movement or items that follow each other in a particular order

default increment is +1 or -1

seq(from=2 , to = 4) // 2 3 4
seq (from=4 , to = 2) // 4 3 2
seq (from =10 , to =20 , by = 2) // 10 12 14 16 18 20
can also be fractional value
seq (to=10 , length=10) // 1 2 3 4 5 6 7 8 9 10
seq (from=1 , length=5) // 1 2 3 4 5

lecture-26 1.23 : 10 // 1.23 2.23 3.23 ..... 9.23
1.23 : 10.54 // 1.23 2.23 3.23 ..... 9.23 10.23
10.54 : 2.23 // 10.54 9.54 8.54 .... 3.54 2.54
-1.23 : -10 // -1.23 -2.23 ..... -8.23 -9.23
-5.23 : 6 // -5.23 -4.23 ... -1.23 .. -0.23 0.77 1.77 2.77 3.7 4.77 5.

Index vector x = c(9,8,7,6)
ind = seq(along = x) // ind = 1 2 3 4

Generating current date and time

Sys.time() provides current time and date from the computer system 21:23:57 IST"

Sys.Date() // "2023-11-09"

lecture-27 Sequences of dates and alphabets

from → req arg to (optional)

> seq (as.Date("2010-01-01"), as.Date("2017-01-01"), by = "year")

// "2010-01-01" "2011-01-01" "2012-01-01" ... "2017-01-01"

> letters // lowercase alphabets "a" "b" ..... "x" "y" "z"

> letters[1:3] // "a" "b" "c"

> letters[2] // "b"

> LETTERS // uppercase

lecture-28 Repeats

rep() - replicates numeric values, or text, or values of a vector for specific no. of times

rep() - cmd to replicate values in vector

rep(x, times=3) // if x = 1 2 → 1 2 1 2 1 2

rep(x, each=3) // 1 1 1 2 2 2

rep(1:2, each=2, times=3) // 1 1 2 2 1 1 2 2 1 1 2 2

rep(1:4, 2:5) // 1 1 2 2 2 3 3 3 3 4 4 4 4 4

x = 1 2
3 4

rep(x, 3) // 1 3 2 4 1 3 2 4 1 3 2 4

rep(3:6, 3) // 3 4 5 6 3 4 5 6 3 4 5 6

rep(c("a", "b", "c"), 2) // "a" "b" "c" "a" "b" "c"

rep(2, length.out=5) // 2 2 2 2 2

rep(c(2,3), length=5) // 2 3 2 3 2

rep(c(2,3,4), length=5) // 2 3 4 2 3

rep(c("a", "b", "c"), length=2) // "a" "b"

l-29

Sorting sort(x, decreasing = FALSE, ...)

na.last → for controlling the treatment of NAs

If TRUE, missing values in the data are put last

FALSE, put first

NA, removed

y = c(8,5,7,6)

> sort(y) // 5 6 7 8

> sort(y, decreasing = TRUE) // 8 7 6 5

order

y = c(9,8,5,7,6)
1 2 3 4 5

order(y) // 3 5 4 2 1

order(y, decreasing = TRUE)
// 1 2 4 5 3

mode() → indicates how the object is stored in memory (as

✓ number
✓ character string
✓ list of ptr to other objects
✓ function etc

mode(2.432) // "numeric"
mode(c(3,4,5,6,7,8)) // "numeric"
mode("India") // "character"
mode(print) // function
mode(list(" India", "USA")) // list
mode(factor(c("UP", "MP"))) // numeric

lecture-30 Lists heterogeneous vector-homogeneous

→ lists can be indexed by position // x[[5]] refers to 5th element of x

→ sublist of x → x[c(2,5)]

→ x[["students"]] x$students → element named 'Students

→ 2= [[1]] "water" "juice" "lemonade" ↔ z1[[1]]
[[2]] 1 1 2 2 3 3 4 4 z1[1][2] // NULL
[1] [,1] [,2] z1[[1]][2]// juice
[[3]] 5 6

7 8

lecture-36 print(" ") only one object at a time
try multiple objects → error
soln: cat("The zero", 2*pi);

cat(..., file="", sep=" ", fill=FALSE, labels=NULL, append=FALSE)

fill ↑ how op is broken into successive lines

labels ↑ character vector of labels for the lines printed

append: logical. only used if arg file is name of file. If TRUE, op will be appended to file; otherwise, it'll overwrite contents of file.

formatting

cat(" The sq root of ", x, "is ", format(sqrt(x), digits=3), "\n")
// The sq root of 7 is 2.65 (no quotes)

lecture-37 paste() concatenates several things together
result of paste() can be assigned to a variable in contrast to cat()

paste(..., sep=" ", collapse=NULL)

collapse ↑ values in result are then concatenated into single string with elements being sep by val of colla

paste(1:4) // "1" "2" "3" "4"
paste("I", "love", "Ma") // "I love Ma"

look at ppt lec-37

lec-38 strsplit() splits elements of a character vector

strsplit(x , split , fixed = FALSE, ...)
if TRUE match split exactly , otherwise use regex

```
> x = "The&longest river& is& Nile"
> strsplit( x, split = "&")
 "The"      "longest river "      "is"      "Nile"
> dates = c("2010-07-24", "2021-08-25" , "2022-09-26", "2023-10-27")
> datesplit = strsplit ( dates , "-")
> datesplit     //  [[1]]
                     [1] "2010"   "07"   "24"

                    [[2]]
                     [1] "2021"   "08"   "25"

>datemat = matrix( unlist(datesplit), nrow=4, ncol=3, byrow=TRUE
         [,1]     [,2]   [,3]
   [1,] "2010"   "07"   "24"        as.numeric(      ))
   [2,] "2021"   "08"   "25"
   [3,] "2022"   "09"   "26"        typecast(  )
   [4,] "2023"   "10"   "27"

>strsplit("Hello", split="")        // "H"  "e"  "l"  "l"  "o"
```

lecture-39 nchar(x) //count no.of characters in x
nzchar(x) // T or F returns if x(char vector) is non-em
T non-empty, F empty

```
x = c("A", "B", "C", "")
nzchar(x)    //  TRUE   TRUE   TRUE   FALSE
nchar(x)     //   1      1      1      _idk

tolower(x)   ,  toupper(x)
```

lecture-40 : Operations with strings : substitution

→ sub(old, new, string) : finds 1st instance of old substring within string and replaces it with the new substring

→ gsub(old, new, string) : global substitution , replaces all instances with new

→ grep , grepl → used for searching for matches.
global

grep(pattern, x , ignore.case = FALSE)
if false → case sensitive

value = FALSE
vector containing indices of matches returned

if TRUE
vector containing matching elements themselves is returned.

in middle page.

R

lec-40

```
> str = c("R Course", "exercises", "Include examples of R")
> grep("ex", str, value=T)
   [1] "exercises"     "Include examples of R"
> grep("ex", str, value=F)
  [1]  2  3
```

↑ if value param is not mentioned, default value=F

grepl (pattern, x) → outcome in terms of TRUE or FALSE

```
> str = c("R course","exercises", "Include R ex")
> grepl( "R", str)
  [1] TRUE   FALSE   FALSE
```

Lecture 41 – Data Frames (DF):

c, cbind, vector, matrix, dataframe : combine data

DF: can combine variables of equal length, with each row in the df containing observations of the same unit. to cbind or matrix (but these can't combine

adv: can make changes to data without affecting orig data.

"MASS" package describes fun & datasets to support

painters dataframe is available in MASS

lect-42   Dateframes: Creation and Operations

1st if are are dealing with a data frame

```
> is.data.frame(painters)        // TRUE
```

Create data.frame

```
Ex:    x = 1:16                              # vector
       y = matrix(x, nrow=4, ncol=4)         # 4x4 mat
       z = letters( 1:16)                    # lc alphabets
       df = data.frame(x,y,z)
> df :     X    X1    X2    X3    X4    z
      1    1    1     5     9     13    a
      2    2    2     6     10    14    b
      3    3    3     7     11    15    c
      4    4    4     8     12    16    d
      5    5    1     .     .     .     e
      .    .    2     .     .     .
      .    .    3     .     .     .
      .    .    4     .     .     .
      .    .    .     .     .     .
      .    .    .     8     9     13    m
      14   14   2     6     10    14    n
      15   15   3     7     11    15    o
           16   4     8     12    16    p
```

Structure of the data - str(painters)

// 'data.frame' : 54 obs. of 5 var
$ composition : int 10 15 8 . . . .
$ Drawing : int 8 16 13 . . .
$ color : int
$ Expression : int
$ School : Factor w/ 8. levels "A","B","C", . . . :1111111..

extract a variable from dataframe using $

Ex: painters$School

extract data from df → painters["Da Udine" , "School"]

→load a library - library(MASS)
↑ name of library

→colnames(painters)

→ plot and graphics of data

barplot (table(painters$School))
pie (table (painters$School))

lec-43 → summary (painters $School) returns a detailed freq table for categorical variable.

→generic function used to produce result summaries of results of various model fitting functions

→attach(painters) → var can be refered directly by name

→ summary(School)

→ detach( ) recovers default setting

→subset (painters , condi) : returns subset of a df

~~subs pain~~ subset(painters, School = = 'F') is equivalent to painters[painters [[" School "]] = = "F" ,]

uninteresting columns can be eleminated

> subset(painters, School = = "F", select = c(-3,-5))
↑ 3 & 5 col are not shown

> split (dfname , variable) partitions data set by values of a specific variable.

lecture-44 combining and merging

df : combining

cbind() - combines cols of 2 dfs side by side.

merge() - joining 2 dfs using common column

rbind() - Stacking 2 dfs on top of each other (appending)