Data Science Notes

Contributor: Gurleen

[SGGSCC DU]

Computer Science Notes

DATA SCIENCES using R !!

14th Jan '19

Unit - 1.

chap (1-2) → 5 Marks

- Data Science / Dat Analytics
- Data Mining
- Machine Learning.
- Big Data

Data → Structured (data picked from DB)
↳ unstructured (data picked from any other form other than databases, eg. newspapes cookies, tweets etc)

Analysis:-

Mining: Data → procedure/Algo. → info → processing/model/prediction → pattern

Analysis [eg. 5 yrs Data of elect extract 2018's → prediction of 2019 lapply some predictive modelling → so Mining.

eg. Stores me kon sa product kiske sath rakha hai.

Machine learning :- (i) Supervised learning (under supervision of someone)
(ii) unsupervised learning (Machine is learning by own).

Big Data → large volumes (for now)

Umbrella Activity

Data science → changing data to product.

if making a software

→ which data I have for software ⇒ Analysis.

→

Data Mining subset of Data Science.

STEPS OF DATA SCIENCE.

(i) collection of Data

(jis data ko product me change karna hai, usko collect karna)

through many means → any measures surveys, images. may be textual etc

data can be in csv file.

which is feasible for any particular person.

(ii) Pre-processing Data

Data set MOCK (2015 - 18)

intro to Data Science

Data in form of (structured Data)

| Name | Joining Data | DOB | completion Date | email..

is Data se ~~extract~~ extract karna.

→ Avg. of completion time

predict → how many people are completing the cource → difficult to find.
↳ if I have to find the factor of how many people not completing cource
↳ can be found by no. of days spent for course.

① SELECTION OF IMP. ATTRIBUTES.

② HANDELING MISSING VALUES
↳ jo data specified nahi hai, missing hai to find using any calculation.
Suppose ending date → not given. ←
So, starting date + no. of days spent.
OR,
excluding missing values.
OR use any default value for no place.

(iii) Analysing Data.
↓
Data ~~left~~ would be analysed for further future.

(iv) Deriving insights and generating BI reports.
↓
showing / representation of data
↓
pi chart etc.

(v) Taking decision based on insight.
↓
conclusion finding.
(Warehouse data collect kiya, bread & egg & milk & butter should be kept together.)

Big Data sources
↳ any source can be used to find data.
~~eg.~~
what is your data source?
How is your data?

Recommending System (Amazon for eg.)
↓
increasing revenue by the activity.
→ (eg. Netflix & prime, engaging more by not searching for a movie by watching one)

5 V's of Big Data

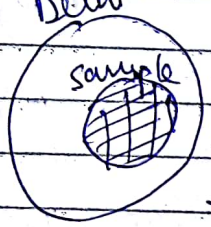
→ Raw data: Volume.
→ Change over time: velocity.

→ Data type: velocity variety.
→ Data quality: veracity
→ info for decision: value making.

Statistics. → Delhi Data Analysis.

→ population & Sample.

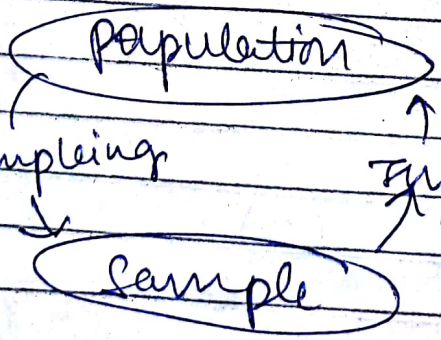
Entire area → Delhi
Sample 1 → North Delhi
↓
predict can't be formed using only this sample.



Sample 2 → North, South, East, West → 200 people each
↳ Better than Previous

Sample 3 → 250 for each part
↳ more Better (total 1000 people)

If any conclusion is valid for given sample & is also valid for Entire area (population)
↓
So, valid conclusion / Statistical inference.



Sampling distribution

Size

Population : N'

sample " = N

"N = ALL"

if sample is valid for populat"

DATA MODELS

P

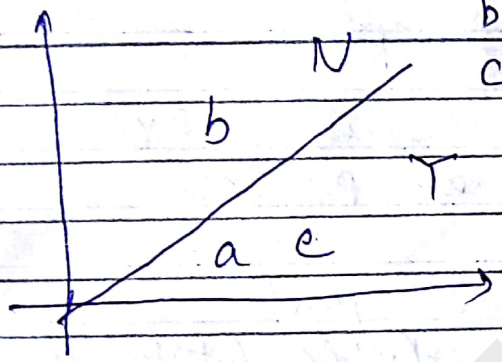
↓

sample

→ classification problem

2 class (Y or N)

	Educated	voted
a	10th	Y
b	12th	N
c	graduate	Y

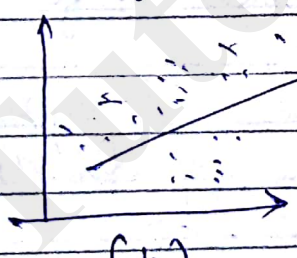


→ Statistical infer the Data (if majority of educated people are lying in Y's)

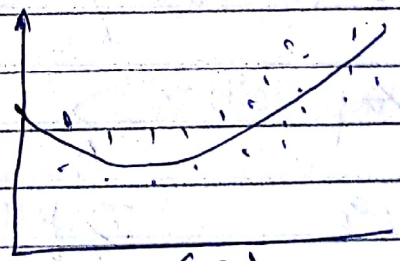
(a) overfitting (estimated 70% study 90%)

(b) under fitting (not hold statistical infer)

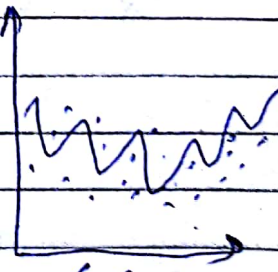
(c) just - fitted value



(b)



(c)



(a)

Exploratory Data Analysis = Data Analysis.

Chap 1 & 2 = End (Unit 1).

15th Jan, 19

Unit - 3 Ref [2] R-Intro (chap 1 - 10)

R → programming lang.
↳ used for data analysis & manipulation

① packages (data analysis, graphics, etc has every other package)

② Support - graphics (only in this technology)

③ case-sensitive

Que 2 Marks
Features of R → Study from Book

① # comments in R
↳ only single line using #

② > prompt symbol
> 1 + 2 ↵
3
>

③ Statement in multiple line
> 1 * ↵ + (automatically puts a +)
> + 3 ↵
3
↓ represent an incomplete statement.

R → untyped language (need not to specify Data types)

Data types in R.

(whatever rules we have for naming a variable, still holds).

① numeric

> a ← 1

(any int, float etc)

> class (a)

↓ tells the datatype of variable a i.e Numeric

output: [1] "numeric"

↳ represent single output.

Assignment operator

=, ←, →

- a = a+1 // mathematically wrong
- a ← a+1
- a → b or b ← a

> b ← 131

② Integer → any no. w/o fractional part

> a ← 1L # L represents integers value

> class(a)

[1] "integers"

> c ← as.Integer(b) # Type casting

> class(c)

[1] "Integers"

③ logical – T/F

```
> v ← TRUE
> class(v)
```

[1] "logical"

• option to save work environment
• .R file

```
> v ✓
TRUE
> print(class(v)) ✓
#True
```

```
> print(class(v))
> print(a)  X
> v  X
```

④ Complex → used to store complex numbers.

```
> a ← 1+i3
> class(a)
```

[1] "complex"

⑤ character

```
> a ← "abc"
> b ← 'abc'
> c ← "123"
```

⑥ Raw
In this datatype, characters are explicitly stored as ASCII characters / codes.

```
> a ← charToRaw("Hello")
> class(a)
```

[1] "Raw"

a will store a value of HELLO concatenated ascii value.

• R-objects

* vectors → collection of certain values, values need to be atomic & of single data-type (like array)
[1, 2, 3] → vector

* Lists → any every need not to be atomic
[[1,2,3], array, matrix]
↳ first element of list is list

* Arrays → 2D Matrix i.e for eg 3×3
* Matrics → 3×3×2 dim 2
↳ 2 Matrix of ~~with~~ 3×3 ~~etc~~
#can have multiple dimension

* Factors → Categorical Data

grade	sex
A	m
B	f
C	m
A	f

→ value is categorised in 2 values i.e m & f.
categorical value = 3 i.e A B C

levels 3 ← A B C

levels = 2 ↓ m f

* DataFrames :- Something like a matrix but every value can have different datatypes.
# Readily used

eg.

Sn.	Name	Gender	Semester
1	:	:	:
2			

• csv file will be stored in DataFrames.

VECTORS

characters

>v ← c('red', 'green', 'blue')

function combines all its arguments

c() # function

Memory

Red	green	blue
0	1	2

indexing same as normal array

> class(v) #

[1] "character"

w/o c(), vector can't be created

Lists

→ list() # function

>L ← list(c(2,1,3), 21.2, 'a')

① vector ② numeric ③ char

> print(L)

```
[[1]]
 [1] 2 1 3
[[2]]
 [1] 21.2
[[3]]
 [1] 'a'
```

* class(L)

find output

Matrix (dim 2)

```
> m ← matrix (c(1,2,3,4), nrow=2, ncol=2, byrow=TRUE)
```

vector with value — no. of rows — no. of columns

dim = nrow × ncol

1	2
3	4

byrow = T (default)

1	3
2	4

By Row = F

```
> m ← matrix(c(1,2,3,4), nrow=2)
```

then ncolumn = 2 (By default)

if value are 9 & nrow=2 ✗ (NO)

ARRAYS

```
> a ← array(c(1,2), dim=c(3,3,2))
```

c(1,2): value of elements

vector of dimension: c(3,3,2) → 3×3×2

No. of elements = 18

mismatch in both arguments. So the values would be repeated

```
> print(a)
> a
```

, , 1

	[1]	[2]	[3]
[1]	1	2	1
[2]	2	1	2
[3]	1	2	1

, , 2

	[1]	[2]	[3]
[1]	2	1	2
[2]	1	2	1
[3]	2	1	2

[x, —] x^th Row [—, x] x^th column

22nd Jan, 2019.

▲ vectors

creating vectors

① using c() function.

```
>x <- c(1,7,6,9);
> x
```

```
[1] 1 7 6 9
```

numeric by default, if you want int, write 1L

~~②~~ typeof(x) # class(x)

```
>y <- c(1, 2, true, "a")
```

logical → numerical → character.

```
>y
```

```
[1] "1" "2" "TRUE" "a"
```

② using infix function ":"

work only on continuous values. i.e.

gap b/w the data members is one

> v ← c(1,2,3 . . . . , 9,10)

≡

> v ← 1:10

> y ← 11:1100 # (from 11 to 1100)

> y ← 2:-2; y # concatenation

≡

> y ← 2:-2
> y

concatenation can be done for any no. of statements

[1] 2 1 0 -1 -2 | Output window

> z ← 3.4 : 10.2; z

[1] 3.4 4.4 5.4 6.4 7.4 8.4 9.4

③ using seq() # stepsize would be given

seq(from = a, to = b, [by = step-value, length.out = len of vectors])

> x ← seq(1,7); x

[1] 1 2 3 . . . 7

default values
★ step-value = 1
★

> x ← 1:7
> x ← c(1,2,3,4,5,6,7)

} same.

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

```
> x ← seq(1, 10, by = 2); x
[1] 1 3 5 7 9

> x ← seq(1, 10, length.out = 10); x
                    ↳ x will have a length of 10
 [1]  1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
```

if given all the 4 value, the funct would'nt work

multiplicity of any value in seq or : is 1.

④ using rep()

```
> f ← rep(0, 5); f
[1] 0 0 0 0 0

> f1 ← seq(1:3, 4)
[1] 1 2 3 1 2 3 1 2 3 1 2 3

> length(f1)
[1] 12

> h ← rep(4:6, 1:3); h
[1] 4 5 5 6 6 6
```

Recycling

```
h1 ← rep(1:10, 1:5)
      ↓10   ↓5   (5 a multiple of 10)
[1] 1 22 333 4444 55555 6 77 88 8 999 10 10 10
```

> h1 ← rep(1:10, 1:4); h
(10, 4 [4 is not a multiple of 10])

so, a warning will occur
some interpreter will give out some not.

[1] 1 22 333 4444 5 6 7 7 7 8 8 8 8
9 10 10

> x ← rep(1:2, each = 2); x

[1] 1 1 2 2

> y ← rep(1:2, time = 2) # y ← rep(1:2, 2)
> y

[1] 1 2 1 2

Indexing a vector

a [index]

① indexing starts with 1.
② Multiple indexed value can be passed
eg. a[c(1,2)] = > a[1]
> a[2]

③ index can be -ve value.

eg.

[1]	[2]	[3]
1	2	3

> a ← c(1, 2, 3)
> a[-1]
↓
all elements except 1.

[1] 2 3

```
for (i=1 ; i<5 ; i++)
    cout << a [i%2 == 0]
```

[i]
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
[i] 2.0
[i] 3.0 3.5 4.0 ~~3.0 3.5 4.0 4.5 5.0~~

①
```
> d <- seq (1,5, by=0.5); d
> d [3]
> d [5:7]
> d >2.8
  ↳ # [1] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
> d[d>2.8] → # [1] 3.0 3.5 4.0 4.5 5.0
```

②
```
t <- c(0,1,2,3,4,5,6)
x <- t[c(2,3,6)]
print (x)
v <- t (c(TRUE, FALSE, FALSE, TRUE))]
print(v)
x <- t [c(-2,-3)]
print (x)
```

```
[1] 1 2 5
[1] 0 3
[1] 0 3 4 5 6
```

- ls() → listing of all the values
- rm(x) → x will be deleted by R-environment
- rm(c(x,y)) → rm(x) rm(y)

R-Studio

script	variables in work environment
console o/p	Packages + library

x ← c(0,1,2)	x

22nd Jan, 19.

> unique(a)
NA #(not available)
↳ same as null.

> x ← c(1, 2, NA, 3)
> length(x)
[4]

Na have a memory location with no value.

NaN → Not a No.
Inf → infinity
-Inf → -∞

vectors are dynamic in R.

```
> x[1] <- 3; x
 [1] 3  2  Na  3
```

```
> x[10]      # No error as like out of Bound
 [1] NA
```

```
> length(x) <- 6; x
 [1] 3  2  Na  3  NA  NA
```

A vector having Na don't properly function for "functions"

eg. `> sum(x)` # 3 + 2 + Na + 3 = 8 + Na = Na

```
 [1] Na
```

```
> sum(x, rm.na = TRUE)
```

✓na.rm (check) = exclude this condition

```
 [1] 6
```

can be used for mean also,
length include Na.

```
> max(x)
 [1] 3
> min(x)
 [1] 1
```

①
```
> x = 1:8
> x[4] = -9; x
> x [c(1,4)]
> x [x>3]
> x [x>3 & x<8]
```

```
x[x %% 2 == 0]

 [1] 1 2 3 -9 5 6 7 8
 [1] 1 2 3 -9
 [1] 5 6 7 8
 [1] 5 6 7 [1] 2 6 8
```

(2)
```
> y ← (1:5)^2 ; y
> y [2:4]
> y [c(-2,-4)]
> y [c(1,-1)]      X Not allowed
> y [-1.9]  = # y[1]   [+ve & -ve indexs mixing is not allowed]
> y [6]     = # Na
> y [1.9]   = # y[1]
> y+1
> y[5:1]
```

(3)
```
> a ← seq(1, 19, by=2); a
> rep(1:3, length.out=9) → b; b
> a ← a[1:4]; a
> a + (2*1:5) →
```

```
> Q ← 2*1:5
[1] 2 4 6 8 10
```

output

(2)
```
[1] 1 4 9 16 25
[1] 4 9 16
[1] 1 9 25
[1] 4 9 16 25
[1] Na
[1] 1
```

functions.

```
abs(x)
sqrt(x)
ceiling(x)
floor(x)
round(x, digit=n)
trunc(x)   # truncate the decimal part.
```

5M

Que Given a vector f as > f ← c(0,1,1,2,3,5,8,13,21,34)
what is the output of the following commands

```
① > f[1:3]                    [1] 0 1 1
② > f[-(1:3)]                 [1] 2 3 5 8 13 21 34
③ f<10                        [1] T T T T T T T F F F
④ f[f<10]                     [1] 0 1 1 2 3 5 8
⑤ f[f%%2==0]                  [1] 0 2 8 34
```

29th Jan, 2019

warning → in matrix(1:5, nrow=3, ncol=3, byrow=TRUE)
data length [5] is not a multiple of the no. of rows [3]

```
> c ← matrix(1:5, nrow=3, ncol=3, byrow=TRUE); c
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    1
[3,]    2    3    4
```

we can rename the dimensions of the matrix

```
> rownames ← c("r1", "r2", "r3")
> columnnames ← c("c1", "c2", "c3")
```

either a list or a vector list()

```
> p<-matrix(c(9:1,3,3,TRUE, dimnames =
  list(row_names, col_names)))p
```

```
> a
     [1] [2] [3]
[1]   1   2   3
[2]   4   5   6
[3]   7   8   9
```

```
> a [2,1]
 [1] 4
```

```
> a [2, ]  # All the column
 [1] 4 5 6
```

```
> a [ , 3]
 [1] 3 6 9
```

```
> a [C(1,2), C(2,3)]
      [ ,1] [ ,2]
[1]     2     3
[2]     5     6
```

```
> a [C(3,2), ]  # All the column.
      [,1) [,2) [,3)
[1]    7    8    9
[2]    4    5    6
```

```
> a [C(3,2)]  # only first column.
```

[1]	
[1,]	7
[2,]	4

> a[-1,]

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

It is possible to index a matrix in a single vector while indexing in such a way it acts like a vector formed by stacking the columns of the matrix one after another. Result is returned as a vector.

> a[1,4] = a[c(1,2,3,4)] → # treated as 1D array.

using logical vectors as index

→ Here rows & columns where the value is true will be returned also, the logical vectors will be recycled if necessary.

→ logical vectors can be mixed with integer vectors for indexing.

3 rows
T, F = recycled when no. are not equal
So, T, F, T
(1st & 3rd row print)

TFF don't work
TRUE & FALSE work.

```
> x
     [,1] [,2] [,3]
[1,]    4    8    3
[2,]    6    0    7
[3,]    1    2    9
```

> x [c (TRUE, FALSE, TRUE), c (TRUE, TRUE, FALSE)]
= x [c(1,3), c(1,2)]

```
     [,1] [,2]
[1,]    4    8
[1,]    1    2
```

> x [c (T, F), c(2,3)]

```
     [,2] [,3]
[1,]    8    3
[3,]    2    9
```

> x [c (TRUE, FALSE)] → # treated as 1D array.

TFTF TF TFT
4 6 1 8 0 2 3 7 9

```
[1] 4 1 0 3 9
```

> x[x>5] # 1D array.

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

> x

```
     [,1] [,2]
[1,]    2    4
[2,]    3   11
```

> det (x)

```
[1] 10
```

function

1. solve → inverse of matrix

> solve (x)

2. sum → calc sum of all elements in the matrix.

> sum (x)

[1] 20

3. product →

> prod (x)

> sum (x[1,])

[1] 5

> y ← diag (3); y

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

> f (x) # transpose

> y ← matrix (1:4, 2, 2); y

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

> x+y > x/y

> x-y > x*y.

element wise add, sub, divide, multiply.

Not matrix wise

> x * y # element wise multiplication not matrix multi.

> x %*% y # matrix multiplication

5M (2 equal given)

Q. write a R script used to solve a system of linear equation.

$$a_1x_1 + b_2x_2 = c_1$$
$$a_2x_1 + b_2x_2 = c_2$$

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & a_2 \end{bmatrix}^{-1} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Step 1: make a matrix using previously mentioned function

Step 2:- make B

Step 3 - y ← solve (A)

Step 4 - ans ← y %*% %*% B OR solve(A,B)

Reading user input

readline → used to read 1 line of input.

n ← as.integer(readline(prompt="Enter n"))

↓ Type casting to integer

↳ same for no. or a string.

more than 1 input (elements).

scan()

```
> x ← scan()↵
1: 3↵
2: 7↵
3: 5
```

ctrl z ≠ ctrl d (for windows & linux)

→ Read 3 input

Que Take 2 vectors of size 2 as an input from the user & combine those vectors to form a matrix where the elements will be filled by row.

funct^n → scan() for input.
→ rbind()

```
> a ← scan()
1: 1
2: 2
3:
```

read 2 items

```
> b ← scan()
1: 3
2: 4
3:
```

read 2 items

```
> x ← rbind(a,b)
```

chap-5
R-Intro

Test for R syntaxes

chap 1, 2, 3 finished

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

06-02-19.

Practicle

users defined functions in R

```
func. name ← function (arg1, arg2...)
{
  =
  =
  =
return # optional, this will not effect the
}                    prototype of a function
```

Prog. 1

```
print ("Fibonacci series")

a ← -1
b ← 1
f ← function (n)
{
while (n>0)
  {
    x ← a+b
    print (x)
    a = b
    b = x
    n ← n-1
  }
}

n ← as.integer (readline ("Enter x"))
f(n)
```

to implement

Ques Write a R script a linear seach using function Take the vector & the key to search as an input from the user.

Input → using scan

12/02/19.

→ name of function

→ code for Armstrong Number

Sum of digits in a cube of no. is equal to no.

```
f ← function (n)
{
  s ← 0
  m ← n
  while (n > 0)
  {
    r = n %%% 10 # modulas returns
    s = s + (r*r*r)
    n = n %/% 10 # Integer division.
  }
  print (m)
  print (s)
  if (m == s) {
    print("a")
  } else {
    print("b")
  }
}
n ← as.integer(...)
f(n)
```

Input n=6

s ← 0
m ← 6
6 * 6 * 6
216
6
216
b

- Arrays in R.
  - ↳ n-dim obj

3 × 4 × (2)
↓
matrix level

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

based on { row column } matrix index.

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

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

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

array (data, dim = c (r, c, m), dimnames = list (m, n, mn))

eg: a → array (1:24, dim = c (3,4,2))
24

eg: v1 ← 1:12
v2 ← 13:24
b ← array (c (v1, v2), dim = c(3,4,2))

All the functions are applicable at this place.

eg:
> v1 ← c (5, 9, 3)
> v2 ← c (10, 11, 12, 13, 14, 15)

> x ← array (c (v1, v2), dim = c(3,3,2))
> print (x)

1st arg has to be data or matrix with a element could also be provided.

, , 1

	[,1]	[,2]	[,3]
[1,]	5	10	13
[2,]	9	11	14
[3,]	3	12	15

, , 2

	[,1]	[,2]	[,3]
[1,]	5	10	13
[2,]	9	11	14
[3,]	3	12	15

eg.

```
> i <- array(c(1:3, 3:1), dim = c(3, 2))
> i

      [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
```

```
> a <- array(1:24, dim = c(3,4,2))
> row.names <- c("r1", "r2", "r3")
> col.names <- c("c1", "c2", "c3")
```

names to a pre-defined array.

```
> m.names <- c("m1", "m2")
> b <- array(1:24, dim = c(3,4,2), dimnames =
        list(row.names, col.names, m.names))
```

output

	c1	c2	c3
r1			
r2			
r3			

functions to check
- → name()
- → attributes()
- → mode()

is.matrix() # object is matrix or not.

```
> a = array(1:9, dim = c(3,4,1))
> a
, , 1
      [,1] [,2] [,3] [,4]
[1,]
[2,]
[3,]
```

→ if this one (1) was not written it then is a matrix

→ works on every function transpose, product etc.

different array not this

a[1,2,1] # 4
a[,2,] # 4 5 6
a[,,] # a entire array.
a[c(1),c(2,3),2] # 13 16
a[c(TRUE,FALSE),c(2,3),2]

a[-1,1,] X
a[-1,,] X

a[-1,,] ✓

→ either the name of array
→ sum(a) # sum of matrix a(,21)
→ nrow(a) } ?? no. of row & col.
→ ncol }
→ mean
→ median
→ %*% # dimensions has to be same

x

	C_1	C_2	C_3
r_1	5	10	13
r_2	9	11	14
r_3	3	12	15

m_1

	C_1	C_2	C_3
r_1	5	10	13
r_2	9	11	14
r_3	3	12	15

m_2

INDEX MATRIX

```
> x <- array(1:20, dim=c(4,5))
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

> i <- array(c(1:3,3:1), dim=c(3,2))
> i
     [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
```

```
> x[i]
[1] 9 6 3

> x[i] ← 0.
> x
```

negative indexes are not allowed in matrix. index.

NA & '0' values are allowed.
row in index matrix containing 0 is ignored.

```
   [,1] [,2] [,3] [,4] [,5]
[1,]          0
[2,]     0
[3,]  0
[4,]
```

LISTS

difference bet" vector & list

① vector → homogeneous { 1D.
list → heterogeneous]

→ you can't have vector inside a vector.

```
v ← c(1, 2, c(1,2)) ≠
v ← c(1, 2, 1, 2)
```

② vector → can't be nested
list → can be nested.

```
> list_1 ← list("Red", "G", c(21, 32, 11), 51.23, TRUE)
> list_1
```

↓ heterogeneous in nature

```
[[1]] # components of lists
[1] "Red"
[[2]]
[1] "G"
```

```
[[3]]
 [1] 21 32 11

[[4]] 51.23
 [1] 51.23

[[5]]
 [1] TRUE.
```

Suppose a matrix with 100 elements of the screen size is 20.

```
    >V
[1]  1 2 ..... 20
[21] 21, 22 23 24 .. 40
[41] 41, 42 ..... 60
```

corresponding to the object's element being printed on the line.

```
> list_1 [1]
[[1]]
 [1] "Red"
> list_1 [[1]]
 [1] "Red"
```

* in vectors, if size = 6, we give 8th element. So 7th is NA.
* in list, it is NULL.

```
> length( list_1)
[1] 6
```

```
> list_1 [[8]] = c(1, 1)
> length (list_1)
[1] 8
```

```
> list_1
$
:
:
[[7]]
 [1] NULL
[[8]]
 [1] 1 1
```

```
> list_data ← list(c("Jan","Jeb","M"),
                   matrix(c(3,9,5,-1,2,8),
                   nrow = 2),
                   list("green",12.3))
> names(list_data) ← c("1st","A","B")
> list_data.
$'1st'
[1] "Jan" "Jeb" "M"

$A
     [,1] [,2] [,3]
[1,]    3    5   +2
[2,]    9   -1    8

$'B'
$'B'[[1]]
[1] "green"

$'B'[[2]]
[1] 12.3
```

```
> x ← list(day = "Tues", lecture = "DS", date =
                                  c(13,2,2018))
> x
$day
[1] "Tues"
$lecture
[1] "DS"

$date
[1] 13 2 2018
```



```
> x $ day
[1] "Tues"

> x $ date
[1] 13  2  2018
```

Merging list

```
> is.list(x)
[1] TRUE

> y = c(1)
> is.list(y)
[1] FALSE

> y = as.list(y)
> is.list(y)

[1] TRUE

> new_list = c(x,y)
> new_list

$day
[1] "Tues"
$lecture
[1] "DS"
$date
[1] 13  2  2018
[[4]] [1] 1
```

- c function is also used to combine multiple lists

x & y list are getting combined into new list.

c(x, y) ≠ c(y, x).

Converting list into vector

sum, prod etc functions are not applicable in list. So we have to convert it to vectors.

use unlist() function.

```
> list1 ← list(1:5)
> list2 ← list(10:14)
> list1 + list2
   Error
> v1 ← unlist(list1)
> v2 ← unlist(list2)
> v1 + v2
[1] 11 13 15 17 19
```

12th Mar, 19.

```
> class (data)        # numeric
> f clars (f1)
[1] factor

> levels (f1)         # will return the levels
                      corresponding to factor f1
[1] "East" "North" "West
```

changing the order of the levels

factor → ordered factor
→ unordered d
(by default unordered)

> max (f1) or min (f1)
will give max^m & min^m level of f1
If no use when ordered.
or max
[1] min is not meaningful for factors

> is.ordered (f1)
[1] FALSE

> f2 ← factor (data,
levels("east", "north", "west"),
ordered = TRUE)

Ordered ke case me,
necessary to specify
levels

> f2
[1] ———
Levels: east < north < west

> max (f2)
[1] "west"

> str (f1) # structure of f1
Factor w/3 levels 1 3 1 2 3

e — 1
w — 3
e
n — 2
w

Output Questions

```
> f1[3]                          [1] east
> f1[c(2,4)]                     [1] west north
> f1[-1]                         [1] west east north west
> f1[c(TRUE, FALSE)]             [1] east east west
> length(f1)                     [1] 5
> f1[6] <- "South"   # since no level is south so, invalid
invalid factor level, NA generated
> f1[2] <- "south"   # 2 ki jagah NA.
> f1
[1] ,  ,  ,  ,  , NA
levels: "east" "north" "west"

# Adding levels to an existing factor
> levels(f1) <- c(levels(f1), "south")
> nlevels(f1)                    [1] 4
> f1[6] <- "south"   # now valid.
```

applications of visualization of data

tapply()

This funct^n is used to sum a pre-defined or a user defined funct^n on a factor

employee	region	incomes
e1	e	20k
e2	south	30k
	n	
	w	

→ can be treated as factors (we want east ki avg salary)

~~> mean(income)~~

> region ← c("e", "w", "e", "n", "s", "e", "n", "s")

> r1 ← factor(region)

> incomes ← c(20, 40, 60, 80, 30, 30, 60, 40)

> mean_income ← tapply(income, r1, mean)

(factor; income → data vector (same calc. jisme karne hai); mean → funct^n that you want to sum (can be user defind or pre-defined))

> mean_income

e	n	s	w
20	..	..	..

2018

Qu differentiate bet ordered & unordered factor and OUTPUT Question.

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well

str as factor = TRUE by default

items ← data.frames (colour = c('red', 'blue', b, x, g, g, y, b, g, y, s, b, b, g, s, y)
size = c(5, 10, 11, 6, 15, 16, 20, 9, 13, 18, 7, 14, 8, 13, 6, 10))

> tapply (items$size, items$colour, sum)

blue	green	red	yellow
52	57	24	56

> tapply (items$size, items$colour, mean)

27th Mar, 19

unit-5 (chap-10 GRAPHICS)
(5-10 Marks)

> data() # whatever data we have in R- environment
> data(iris) # load iris dataset in current environment.
> install package...
> str(iris) # structure of dataset.
> summary(iris)
> head(iris) # 1st 5 records.
> head(iris, n=10)
> tail(iris) # last 5

head always return 6 rows but tail wise 5 row

```
> tail (iris, -6)
> head (pressure)
> max (pressure$temp)
> min      "      "
> mean     "      "
> median   "      "
> quantile "      "
> summary  "      "

> nrow (iris)
> ncol ((iris)
> dimension (iris)
```

	data sets divided into
quatile	4 parts
decile	10 parts
percentile	100 parts.

Graphs.
- scatter plot graph
- line plot graph
- bar plot graph
- box plot grap # statistical
- histogram # like bar plot.
- pie chart

SYNTAX

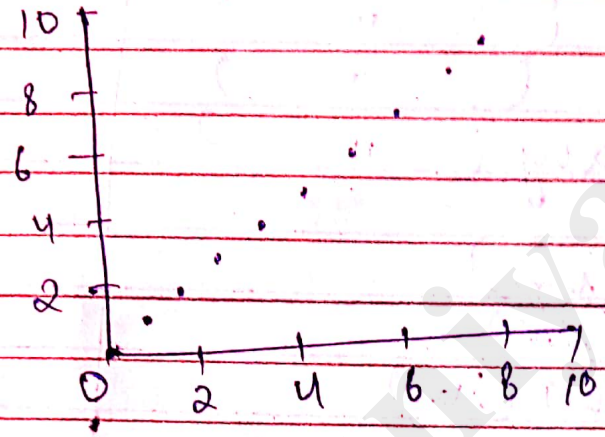
① plot(X, Y, main, xlab, ylab, xlim, ylim, axes)
(title) (table)
(heading)

plot(X → mandatory parameter.

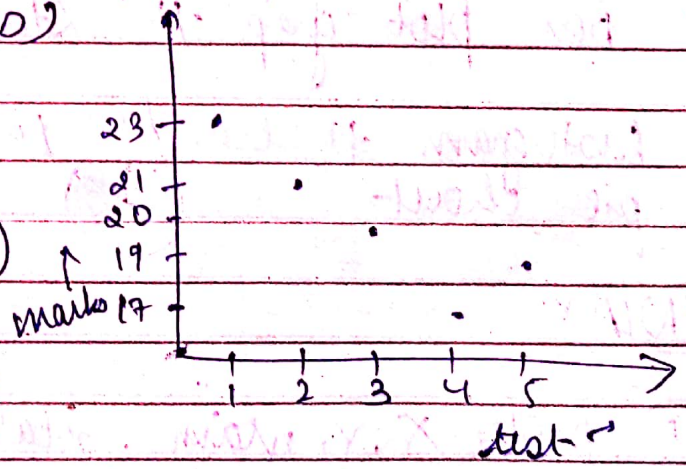
main = "title"
xlab = "speed"
ylab = "distance"

xlim = c(5,50) # limits
ylim = c(0,20)
axes → by default TRUE
is false if you don't want to show axes.

```
> x <- 1:10
> plot(x)
```




```
> plot(x, y)
> plot(dfm)  # by default graph of 1st & 2 col. of dataframe
>test <- 1:5
> marks <- c(23,21,20,17,19)
> plot(test, marks)
```



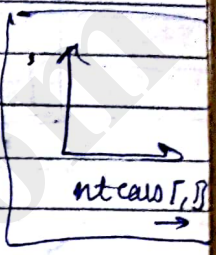

```
> df = data.frame(test, marks)
> plot(df)
```

if 2 other col for graph
> plot(dfc, dfd)

Unit 5 - 15 Marks

> data(ntcars)
> head (ntcars)

> plot (ntcars [, 1], ntcars[, 2]) or
plot (ntcars $ mpg, ntcars $ cgl)



OR

> attach (ntcars) # ntcars is added to the reached path for R.
> plot (mpg, cgl)

> plot.(x,y) = > plot (y ~ x)

> ~~dataframe~~ df = read.csv ("path of file").
> str (df)

> nrow (df)
> ncol (df)
> dim (df)

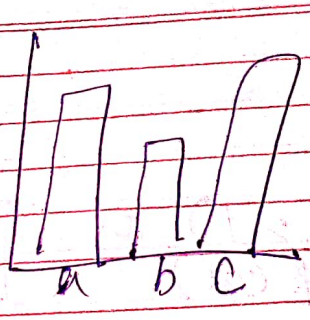
> plot (df $ length, df $ weight, main = "length vs weight",
xlab = "length", ylab = "weight")

BAR CHART

Syntax:

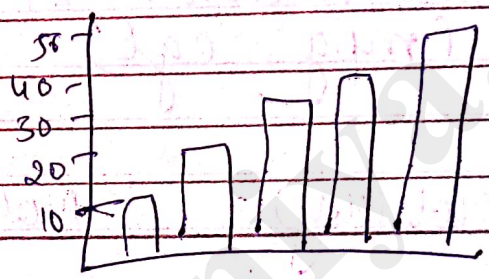
barplot (H, xlab, ylab, main, names, arg, ... col)

- H → vector or matrix
- xlab, ylab → label
- main → heading
- names → vector of names appearing under each bar
- col → colors for bars



vectors would be recycled for names & colors

```
> v <- c(10, 20, 30, 40, 50)
> barplot(v)
```



Eg: # plot a barchart of a mean temperature by months using airQuality dataset

```
> data(airquality) # to load dataset
> heights <- tapply(airquality$temp, airquality$month, mean)
```

```
> tapply(attribute (jisme apply hoga), attribute (factor, jisse divide hoga), mean (function))
> heights
> barplot(heights)
> barplot(heights, main = "mean temp. by month",
          names.arg = c("may", "june", ...),
          xlab = "month", ylab = "temp")
```

Shades change according to size of bar.

> gray → [limits → should be 0 to 1.]
> length (heights)
> rank (heights) ## rank is using order

5	6	7	8	9
1	3	4	5	2
(value chosi)				(Back value)

> table (heights)

> rel.hts <— rank (heights) / length (heights)

> rel.hts

5	6	7	8	9
0.2	0.6	0.8	1.0	0.4

> grays <— gray (1 — rel.hts) ## (rel.hts) same o/p

> grays

> barplot (heights, col = grays,

last year

Que write a code in R to plot a graph to depict the relation b/n temperature in celcius (x axis) & temp in Fern (y-axis) using the formula

$$T(F) = T(C) * \frac{9}{5} + 32$$

Label the axis & give the title as "C vs F" to your graph

5 Marks

Soln :- values should be given by person.

Que Store the following data in a dataframe & perform the following.

A	B
alpha	100
beta	120
gama	80
delta	110

(i) display the contents of dataframe.
(ii) draw a barplot & name the bars acc. to the values of column A.

28th Mar, 19

Line Plot

syntax:

```
>plot (x, y, type = "l")
```

↓ same plot function with another attribute = "l"

eg. plot (df, type = "l")

Box Plot :- statistically representing data

→ mean (avg)
→ std. deviation ($\Sigma (x - \bar{x})^2$)
→ max & min (Range)
→ outlier (temp limit = 30, we are getting 60° @ some place ↓ remove outliers)
→ quatile ↓ dividing into 4 parts.

syntax: boxplot(x, data, notch, varwidth, names, main)

outliers
max
③ median
② median
① median
min

x → vector (any column)
data → dataframe
notch → logical value either true or false
varwidth → width manipulation of box
names → names for boxes (like names.arg)

eg:
boxplot(pressure$temperature)

```
> data(mtcars)
> str(mtcars)
> head(mtcars)
> input <- mtcars[,c('mpg','cyl')]
> head(input)
> dim(input)
> dim(mtcars)
> boxplot(mpg)
> summary(mpg)

> boxplot(mpg, cyl)
> boxplot(mpg ~ cyl.)   > boxplot(cyl, mpg)
```

if 1st arg is factor then no. of boxplot = no. of levels of 1st attribute

cyl has 3 levels, so 3 boxplot (corresponding to value of mpg)

Book eg. (check)

> table (input $ cyl)

4 6 8 # 4 ke corresponding 11 rows.
11 7 14

Histogram

→ frequency related.

syntax:-

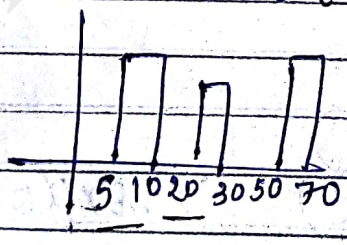
hist (v, main, xlab, xlim, ylim, breaks, col, border)

★ histogram can only have 1 column or attribute. Since we are talking about frequency.

v → vector
main → title
col → colors
border → for border color
border = c("Red", ..)

xlab → label
ylab is frequency by default
xlim & ylim → limits

breaks → mention the width of each bar



eg:

> data (cars93)
> dim (cars93)
> names (cars93)

> hist (Cars93$MPG.City)

Que Read the given text file & perform the followi

(a) draw a histogram showing the occurances of various grades.

(b) Boxplot of variable "final" wrt grade
boxplot (final, grade)

Sec 2.6

(1) mean (x)
(2) median (x)
(3) sd (x) # standard deviation
(4) var (x) # variance
(5) cor (x, y) # corelation
(6) cov (x, y) #

1st April, 19

Que Read the given Table in a dataframe & answer the following questions 6 Mark

empid	Salary (Rs)	Experience (yrs)	Age (yrs)	Category
E1	60,000	11	48	married
E2	15,500	2	30	unmarried
E3	45,00	7	35	M
E4	1,59 00	15	47	M
E5	60,000	11	47	M
E6	15,500	2	28	Unmarried

(i) draw a boxplot for experience age of an empl label the axes of the graph appropriately &

use different colors for different boxplots

(ii) draw a histogram for category

(iii) draw a barchart of the experience of employees showing emp. Id as label for bars.

syntax to read a table

(a.txt) → path.

```
df ← read.table("__/a.txt", header=TRUE)
```

if want to make data frame

```
df ← data.frame(...)
```

Que 2 :- consider the given data, & answer following

Item_Id	Length	Category
i1	65	A
i2	66	B
i3	71	C
i4	70	D
i5	50	A
i6	51	B
i7	40	C
i8	30	D

(i) write a command to display only the last 4 rows of the given dataframes.

(ii) display only rows of the DF where the category is C.

(iii) draw a barplot to depict the mean length of all the items by their categories. Label the axis & graph appropriately.

questions for theory

(iv) draw the boxplot showing the statistical summary of all the items wrt their categories.

(v) write a command to display maximum length within the dataframe.

Ques3 write commands to perform the following.

(i) load iris dataset.
(ii) find the dimension of the structure of the dataset.
(iii) print first seven rows of the dataset.
(iv) write a command to display the individual count of all the species present in iris dataset.
command → Table
(v) write a command to display the names of all the attributes int in the dataset.
(vi) draw a scatterplot of width & length
sepal.width & sepal.length.
(vii) analyse the output of the command
plot(iris)
(viii) draw a boxplot indicating the statistical aspects of all the species wrt sepal.width

See commands & check on these datasets also
① mtcars
② airquality.

Section 2.6

> x ← c(0,1,1,2,3,5,8,13,21,34,NA)

[Standard, variation / deviation] → commad syntax & output.

> mean(x) # if na.rm = TRUE

(8.8) ↓

$$\# \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad ; n \rightarrow \text{length}$$

> median(x) # n → odd / even

(4) odd: $\left(\frac{n+1}{2}\right)$ even: $\frac{\frac{n}{2}^{th} + \frac{n}{2}+1^{th}}{2}$

> var(x) # $\frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2$

> sd(x) # $\sqrt{var(x)}$

$$var(x) = \frac{1}{10-1} \sum_{i=1}^{10} (x_i - \bar{x})^2$$

$$= \frac{1}{9} [(0-8.8)^2 + (1-8.8)^2 + (2-8.8)^2 + (3-8.8)^2 + (5-8.8)^2 + (8-8.8)^2 + (13-8.8)^2 + (21-8.8)^2 + (34-8.8)^2]$$

$$= 121.733$$

$$= \frac{1}{9}[$$

$$sd(x) = 11.033$$

Que:- consider the 2 vectors defined

x ← seq(1, 10, by = 2)

(i) find mean(x) & median(x)

(ii) x[6] ← 2
x[7] ← 4
x[c(8,9,10)] ← 2
> print(length(x))
final Mode of x

mode → value which occurs of most of the time but mode(x) doesn't find the frequency & for mode, we have to find a logic

(iii) find standard deviation & variance of x.

(iv) > x[12] ← NA
Now, find mean(x) → answer = Na
Also state the effect of the introduction of Na value on this vector x.

Rectify the syntax so as to calculate the actual mean of vector x

Sol^n:- x(1, 3, 5, 7, 9)

x(1,3,5,7,9,2,4,2,2,2)

(i) > mean(x) (Ri) print (length(x))
[1] 5 [1] 10
> median(x)
[1] 5

(iii) var(x) = $\frac{1}{9}$(77) = 6.67
sd(x) = 2.58

(iv) mean(x) [1] Na

(iv) x (——— , 2/9 , 2/10 , 2/11 , NA , NA/12)

NA.rm = TRUE

Finding mode of x.

where, x = c(1, 3, 5, 7, 9, 2, 4, 2, 2, 2)

(2) > ans ← y[which.max(tabulate(match(x, y)))]

(1) > y ← unique(x); y

① unique funct^n returns a row vector that contains the sorted set of unique values of its arguments

[set → no duplicate values]

> y ← unique(x); y

[1] 1 2 3 4 5 7 9

some versions of R returns a set of unique values w/o sorting it.

[1] 1 3 5 7 9 2 4

[1] [2] [3] [4] [5] [6] [7]

② Match funct^n returns the 1st occurence of the 1st argument in the 2nd argument

> match(x, y)

[1] 1 2 3 4 5 6 7 6 6 6

- Tabulate funct^n takes an integer valued vector & counts the no. of times each integer occurs in it.

```
> tabulate (match(x, y))
1 1 1 1 1 4 1
```

- ~~which~~ max → max value ; (4)
  ans ← y[which.4]
- "which" funct^n returns the index of the logical object when it is true.

ERROR

(v) pairs of same quetion.

y ← rep (1+x, each=2)

find correlation of (x & y) & co-variance of x & y.

cor (x, y) & cov(x, y)

(vi) Remove Na value from x & create a vector xx from x with values less than 4. 0

(vii) y1 is set as 1:6. Now, find correlation of (x1, y1) & covariance of (x1, y1)

FORMULAE

$$cov(x,y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

& $$cor(x,y) = \frac{cov(x,y)}{var(x) \cdot var(y)}$$ ✱ Check

where x & y have to be vectors of same dimensions

(v)

x → 1 3 5 7 9 2 4 22 Na Na

rep 2 2 4 4

→ x ← c(1:3, x, 1:3)

y ← rep (x, each = 2) 1 1 2 2 3 3

or

y ← rep (x, times = 2) 1 2 3 1 2 3

→ Mismatch in dimensions in the vectors x & y.

Soln :-

(v) y (2 2 4 4 6 6 8 8 10 10 3 3
5 5 3 3 3 3 3 3 Na Na Na
Na).

Error

(vi) x ← x[!is.na(x)]; x

[1] 1 3 5 7 9 2 4 2 2 2

x_1 ← x[x < 4]; x_1

[1] 1 3 2 2 2 2 $\bar{x}_1 = 2$

(vii) y_1 ← c[1:6] [1 2 3 4 5 6] $\bar{y}_1 = 3.5$

$$cov = (x_1, y_1) = \frac{(-1)(-2.5) + (1)(-1.5)}{5}$$

$$= \frac{2.5 - 1.5}{5} = \frac{1}{5} = 0.2.$$

$$cor = \frac{0.2}{var(x_1)\, var(y_1)} = \frac{0.2}{(0.2)(3.5)} = \frac{0.2}{0.7} = \times$$

$$var(x_1) = \frac{1}{5}\left[(-1)^2 + (1)^2\right] = \frac{1}{5} = 0.2$$

$$var(y_1) = \frac{1}{5}\left[(-2.5)^2 + (-1.5)^2 + (-0.5)^2 + (0.5)^2 + (1.5)^2 + (2.5)^2\right]$$

×

$$cor(x, y) = 0.169$$

- mean & standard deviation of a dataframe works by column & these funct will only work for the numerical column of data frame

```
> df
```

small	medium	large
:	:	:

① mean(df)

small	medium	large
mean of col1	mean of col2	mean of col3

② ~~med~~ sd(df)

③ median(df) ✗ (wrong)

median(df$small)
median(df$medium)
median(df$large)
mode → no funct, do same as median

|| or tapply & lapply

- variance & co-variance returns a covariance matrix for the dataframes

① var(df)
② cov(df)

	small	medium	large
small	cov(s,s)	cov(m,s)	cov(l,s)
medium	cov(s,m)	cov(m,m)	cov(l,m)
large	cov(s,l)	cov(m,l)	cov(l,l)

TutorialsDuniya.com

Download FREE Computer Science Notes, Programs, Projects, Books PDF for any university student of BCA, MCA, B.Sc, B.Tech CSE, M.Sc, M.Tech at https://www.tutorialsduniya.com

- Algorithms Notes
- Artificial Intelligence
- Android Programming
- C & C++ Programming
- Combinatorial Optimization
- Computer Graphics
- Computer Networks
- Computer System Architecture
- DBMS & SQL Notes
- Data Analysis & Visualization
- Data Mining
- Data Science
- Data Structures
- Deep Learning
- Digital Image Processing
- Discrete Mathematics
- Information Security
- Internet Technologies
- Java Programming
- JavaScript & jQuery
- Machine Learning
- Microprocessor
- Operating System
- Operational Research
- PHP Notes
- Python Programming
- R Programming
- Software Engineering
- System Programming
- Theory of Computation
- Unix Network Programming
- Web Design & Development

Please Share these Notes with your Friends as well