R Tutorial :-

Basic computations :

1. Calculator-

For just practice run the following in R -

(5+9)*5/80-23

sin(100)

exp(5)

log(9), returns the natural logarithm of 9

log10(2), returns logarithm of 2 to the base 10

abs(-94) , abs gives the abosolute value

sqrt(81)

54%%9 , "y%%x" gives the remainder of y/x, here returns 6

ceiling(4.5) , ceiling function round ups to the next larger integer, here it returns 5

floor(4.5), floor function rounds down to next smaller integer, here it returns 4

trunc(5.9), returns the integer part of the number, here returns 5

round(4.2689,digits=3), returns the given number rounded up to the 3 digits, here returns 4.27

2. Variables:

We can assign a variable to any vector as given below:

x=c(4,5,4,9,8,6,9,3)

ans.- [1] 4 5 4 9 8 6 9 3

Run the follwing:-

x=c(4,2,8,9,7)

length(x), gives the number of elements in the vector.

x[1: 3], returns the elements in the position starting from 1 to 3, here find if it is giving the answer correctly.

x[x>4], returns the elements of x which are greater than 4(check)

which(x>4), returns the position of the elements of x which are greater than 4.

3.Generating sequences :-

seq(x,y,by=z) returns a sequence starting with value x, ending at y with increment z.

Perform seq(1,40,3), seq(1,40,5), seq(10), seq(1:50) and check what it gives.

Run seq(1:50,5). What do you get? Now do you understand what is seq(x : y) actually gives?

Perform seq(1:50),seq(40:50),seq(39:50),seq(38:50),seq(21:50). You will better understand now.

seq(x : y) returns a sequence starting from 1 with length abs(y-x+1).

Perform seq(19:50) and seq(50:19). Do you get it?

rep(x,z) returns the value x repeated z times. Run rep(4,6), rep(4.23,50).

Run the following -

x=seq(1:3)

y=rep(x,5)

z=rep(x,each=4)

See what it returns. Do you understand?

4. Accumulating data :-

run the following in R -

x=c(4,25,1,8,90,6,54,75,6)

x1=4*x/5+log(x)

x2=sin(x)+cos(x1)

x3= x2%%2-abs(x2-x1)
y=data.frame(x,x1,x2,x3)
y
Do you get all the variables and there corresponding values accumulated in a table format?

Some more basic commands and basic stats:-

Run the following in console:-
x=c(45,52,65,24,89,75,64,23,56,51,57,58,62,60,67,95,34)
x
sum(x)
mean(x)
sum(x[x>mean(x)])
sum(x[x%%2==0])
var(x) , returns variance of x
sd(x) , returns standard deviation of x
sort(x), returns sorted array of x in increasing order.
min(x)
max(x)
quantile(x)
IQR(x), returnns inter quartile range of x
scale(x) , returns standardised x, i.e., (x-mean(x))/sd(x).
summary(x)
You get it. Isn't it right?
You can perform any function you want. Just for learning purpose play with commands.

Graph plotting:-

plot(v,z) , returns a scatterplot of z vs v,z on y-axis and v on x-axis.
There are many arguments and parameters which can be used inside and outside plot to improve the graph. Run the follwing:-
x=seq(-pi,pi,pi/10)
y=sin(x)
plot(x,y)
now run- **plot(x,y,type='b',main='sine curve')**
see the difference?
Now find the function of **type** and **main** in the plotting(what they do in the plotting).
Search **par** and read it for once. Try to understand the parameters which can be used.
Use of par in plotting:-
run the following-
x=seq(1:30)
y=x*sin(x)
plot(x,y,type='b',xaxt='n',main='sine curve')
axis(1,at=x, labels=x, las=2)
Find in **par** what work '**axis**', '**xaxt**' and '**las**' do.
Run the following:-
illiteracy.1991=c(1010456,3948251,1636908,1154360,36833351,18857757,14366758,10221940,1788537)
illiteracy.2001=c(826595,3636084,1405351,1329783,36504260,21169173,13234798,959564

0,1770311)
states=c("Himachal Pradesh","Punjab","Uttaranchal","Delhi","Uttar pradesh","Bihar","West Bengal","Karnataka","Kerala")
percent.decrease=((illiteracy.91-illiteracy.01)*100)/illiteracy.91
percent.decrease
barplot(percent.decrease, names.arg=states)
Now search and read **barplot**. Try to improve the graph by giving title to the graph, x and y labels etc.
Run the following:-
years=seq(1977,1981)
arts=c(1783,2057,2096,2252,2267)
com=c(590,650,723,848,900)
sci=c(391,478,438,394,554)
law=c(400,552,640,689,926)
subjects=rbind(arts,com,sci,law)
subjects
barplot(subjects,beside=FALSE,legend=rownames(subjects),
main=" Number of students in differnt faculties over years",names.arg=years,ylim=c(0,3500),xlab="Years",ylab="number of students")
Find out the function of **beside,legend** used in barplot.
Check what happens when you use **beside=FALSE.**
Run the following:
years=c(1951,1957,1962,1967,1972,1977,1980,1984,1989,1991,1996,1999,2004,2009)
westbengal=c(7613,10440,10038,13370,13667,15133,21035,25905,32200,31761,37677,37134,37221,42285)
maharashtra=c(11528,16760,11721,14391,14391,17404,19018,22451,28256,23708,28979,32096,34263,36963)
rajasthan=c(3626,4649,5415,7095,7158,8673,9709,11465,14594,12596,13188,17927,17346,17906)
karnataka=c(2824,5798,6733,8044,7917,10596,11289,13857,19320,15807,19155,21488,25139,24544)
voters=cbind(westbengal,maharashtra,rajasthan,karnataka)
matplot(years,voters,type="l",lty=c(1,3,5,6),main="number of voters for some states of India over 1951-2009")
legend("topleft",legend=c("westbengal","maharashtra","rajasthan","karnataka"),lty=c(1,3,5,6), bty='n', cex=.75)

Read **matplot** in help.

Linear Algebra:-

Run the following commands:-
A=matrix(c(1,5,-4,2,-1,0),nrow=2,ncol=3,byrow=TRUE)
A
Find out why we use byrow=TRUE (search help matrix).
B=matrix(c(6,0,7,-3,-2,2),nrow=2,ncol=3,byrow=TRUE)
B
C=matrix(c(1,4,3,-5,-2,1,3,9,-2),nrow=3,ncol=3,byrow=TRUE)
C

check what happens when you perform the following commands:-
C[1,] (does it return the 1st row of C?)
B[,2] (does it return the 2nd row of B?)
D=C[-1,-3]
D (does it returns C by deleting it's 1st row and 3rd column?)
2*A+B
5*A-3*B
diag(5)
diag(c(2,5,4,7))
C%*%diag(3). What do you get? ***%*% is used for matrix multiplications.***
E=C%*%C+5*diag(3)
E
F=A%*%C
F
nrow(F)
ncol(F)
t(C)
M=(C+t(C))/2 (is it a symmetric matrix?)
N=(C-t(C))/2 (is it a skew symmetric matrix?)
det(C)
rankMatrix(C)
do you get the rank of the matrix C from the above command?
Run-
library(Matrix)
rankMatrix(C)
Now you should get the rank.
eigen(C), returns the eigen values and eigen vectors.

Revisit system of linear equations. Consider the following -
Ax=b.
Implies x= inverse(A)*b
and we know A*inverse(A)=I
Now read **solve** in **search help**.
Run-
solve(A)
Does it give the inverse of A?
Hence to find inverse of any matrix M use **solve(M).**
Solve the following system of linear equations:-
2a+5b-c-d+3e=12
a-b+c+d+e=3
2a+b+3c-4d+2e=19
3a-2b-4c+d-e=-6
a+2b+2c-3d-3e=8
Hint:- write the above in the form of Ax=b and use solve(A,b)

Random number generation:-

Try searching for “how to generate random numbers from a given probability distribution in R” in Google.
Try generating random numbers from :
a) binomial distribution,

b) normal distribution,
c)uniform distribution.
You will get functions like rbinom, dbinom, pbinom, qbinom for binomial. Similarly for other distributions you can find.
You can search it in R studio itself. Read all the functions. In the upcoming exercises it will be needed.

Programs for practice:-

1. Write a program to return the numbers between 50 and 5000 whose square roots are divisible by 2.
2. a. Create a plot of function exp(–x)*cos(6πx);, where x takes on data points over interval (0,1) with increment .5
b. On the same figure, plot functions exp(–x) and –exp(–x) . Give different colors/patterns for each curve.