MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH AND STUDIES

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Practical File

For

7th Semester (Academic Year 2021-2022)

SUBJECT NAME: SIMULATION MODELLING

SUBJECT CODE: BCS-DS-771

Submitted By: -

Student Name: Shrey jain Kumar

Roll No.: 1/18/FET/BCS/092

Branch: FET CSE (N)

Section: 7CSB

Submitted To:-

Faculty Name: Dr. Brijesh

Designation: Professor

Department: CSE

INDEX

11	Find Covariance and Correlation between these parameters. Plot the data set.	
12	Write a program to find the structural stability of the given truss bridge.	
13	Write a program to implement single server.	
14	Write a program for pure pursuit problem using SCI lab.	

Experiment-1

Aim: Introduction to Sci Lab.

Overview:-

Scilab is a free and open-source, cross-platform numerical computational package and a high-level, numerically oriented programming language. It can be used for signal processing, statistical analysis, image enhancement, fluid dynamics simulations, numerical optimization, and modeling, simulation of explicit and implicit dynamical systems and (if the corresponding toolbox is installed) symbolic manipulations

Scilab is a programming language associated with a rich collection of numerical algorithms covering many aspects of scientific computing problems. From the software point of view, Scilab is an interpreted language. This generally allows to get faster development processes, because the user directly accesses to a high-level language, with a rich set of features provided by the library. The Scilab language is meant to be extended so that user-defined data types can be defined with possibly overloaded operations. Scilab users can develop their own module so that they can solve their problems.

How to install Scilab –

Whatever your platform is (i.e. Windows, Linux or Mac), Scilab binaries can be downloaded directly from the Scilab homepage 3 Figure 1: Scilab console under Windows. http://www.scilab.org or from the Download area http://www.scilab.org/download Scilab binaries are provided for both 32 and 64 bits platforms so that it matches the target installation machine. Scilab can also be downloaded in source form, so that you can compile Scilab by yourself and produce your own binary. Compiling Scilab and generating a binary is especially interesting when we want to understand or debug an existing feature, or when we want to add a new feature. To compile Scilab, some prerequisites binary files are necessary, which are also provided in the Download center. Moreover, a Fortran and a C compiler are required. Compiling Scilab is a process which will not be detailed further in this document, because this chapter is mainly devoted to the external behavior of Scilab.

1.1.1Installing Scilab under Windows Scilab is distributed as a Windows binary and an installer is provided so that the installation is really easy. The Scilab console is presented in figure 1. Several comments may be done about this installation process. On Windows, if your machine is based on an Intel processor, the Intel Math Kernel Library (MKL) [4] enables Scilab to perform faster numerical computations. 1.1.2 Installing Scilab under Linux Under Linux, the binary versions are available from Scilab website as .tar.gz files. There is no need for an installation program with Scilab under Linux: simply unzip the file in one target 4 directory. Once done, the binary file is located in /scilab-5.2.0/bin/scilab. When this script is executed, the console immediately appears and looks exactly the same as on Windows. Notice that Scilab is also distributed with the packaging system available with Linux distributions based on Debian (for example, Ubuntu). That installation method is extremely simple and efficient. Nevertheless, it has one little drawback: the version of Scilab packaged for your Linux distribution may not be up-to-date. This is because there is some delay (from several weeks to several months between the availability of an up-to-date version of Scilab under Linux and its release in Linux distributions. For now, Scilab comes on Linux with a linear algebra library which is optimized and guarantees portability. Under Linux, Scilab does not come with a binary version of ATLAS [1], so that linear algebra is a little slower for that platform. 1.1.3 Installing Scilab under Mac OS Under Mac OS, the binary versions are available from Scilab website as a .dmg file. This binary works for Mac OS versions starting from version 10.5. It uses the Mac OS installer, which provides a classical installation process. Scilab is not available on Power PC systems. Scilab version 5.2 for Mac OS comes with a Tcl / Tk library which is disabled for technical reasons. As a consequence, there are some small limitations on the use of Scilab on this platform

Experiment-2

Aim: Study of matrix operation using Sci lab.

Matrix-

In mathematics, a matrix (plural matrices) is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns. For example, the dimension of the matrix below is 2 × 3 (read "two by three"), because there are two rows and three columns:

Types of matrixes

There are several types of matrices, but the most commonly used are:

- Rows Matrix
- Columns Matrix
- Rectangular Matrix
- Square Matrix
- Diagonal Matrix
- Scalar Matrix
- Identity Matrix
- Triangular Matrix
- Null or Zero Matrix
- Transpose of a Matrix

Row Matrix:A matrix is said to be a row matrix if it has only one row.

Column Matrix:A matrix is said to be a column matrix if it has only one column.

Rectangular Matrix:A matrix is said to be rectangular if the number of rows is not equal to the number of columns.

Square Matrix:A matrix is said to be square if the number of rows is equal to the number of columns.

Diagonal Matrix:A square matrix is said to be diagonal if at least one element of principal diagonal is non-zero and all the other elements are zero.

Scalar Matrix:A diagonal matrix is said to be scalar if all of its diagonal elements are the same.

Identity or Unit Matrix:A diagonal matrix is said to be identity if all of its diagonal elements are equal to one, denoted by II.

Triangular Matrix:A square matrix is said to be triangular if all of its elements above the principal diagonal are zero (lower triangular matrix) or all of its elements below the principal diagonal are zero (upper triangular matrix).

Null or Zero Matrixes:A matrix is said to be a null or zero matrix if all of its elements are equal to zero. It is denoted by OO.

1. Creation of matrix

1.1 Create a row scaler
matrix - - - - ❼ a =
[1,2,3]

1.2 Create a vector scaler
matrix - - - - ❼ b = [1 ; 2;
3]

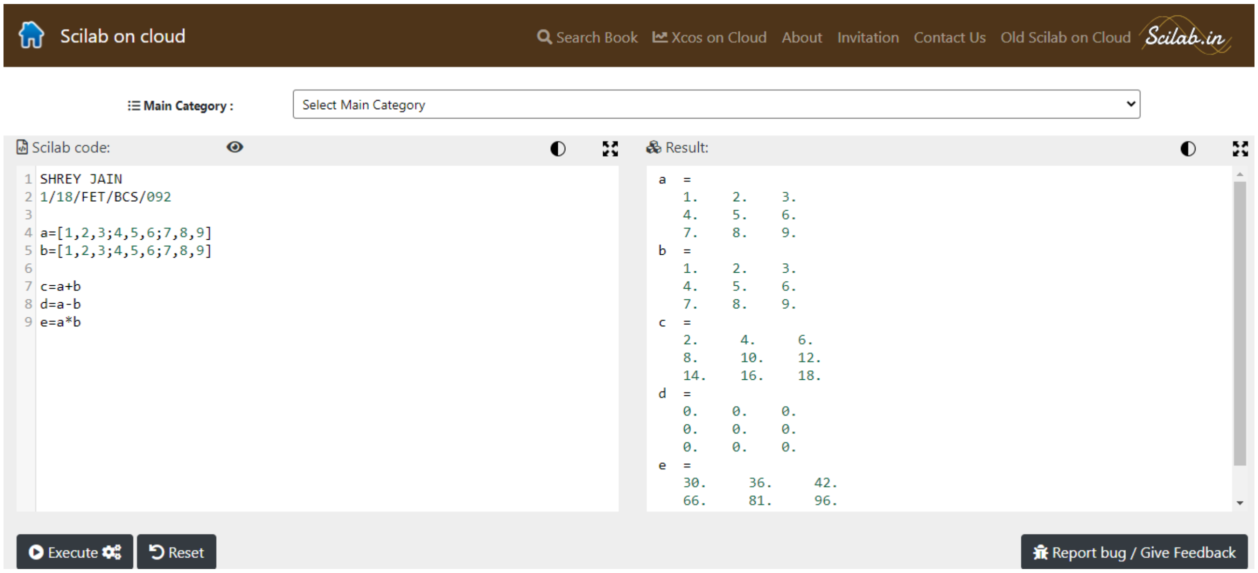
1.3 Create a 3x3 matrix
----❼a=[1,2,3;4,5,6;7,8,9]

1.4 Create a 3x3 matrix
----❼b=[1,2,3;4,5,6;7,8,9]

Mathematical operations on
matrix 1.5 Addition
- - - - ❼ c = a + b

1.6 Subtraction
- - - - ❼ d = a - b

1.7 Multipli
cation - - -
-
❼ e = a * b



```
SHREY JAIN
1/18/FET/BCS/092

a=[1,2,3;4,5,6;7,8,9]
b=[1,2,3;4,5,6;7,8,9]

c=a+b
d=a-b
e=a*b
```

```
 a  =
    1.    2.    3.
    4.    5.    6.
    7.    8.    9.
 b  =
    1.    2.    3.
    4.    5.    6.
    7.    8.    9.
 c  =
    2.     4.     6.
    8.     10.    12.
    14.    16.    18.
 d  =
    0.    0.    0.
    0.    0.    0.
    0.    0.    0.
 e  =
    30.    36.    42.
    66.    81.    96.
```

1.8 Transpose

Suppose AA is a given matrix, then the matrix obtained by interchanging its rows into columns is called the transpose of AA. It is denoted by At.

----→f=(1,2,3)

1.9 Inverse

----❼inv(a)

1.10 Determinant ----❼det(a)

1.11 Eiyen

----❼spec(a)

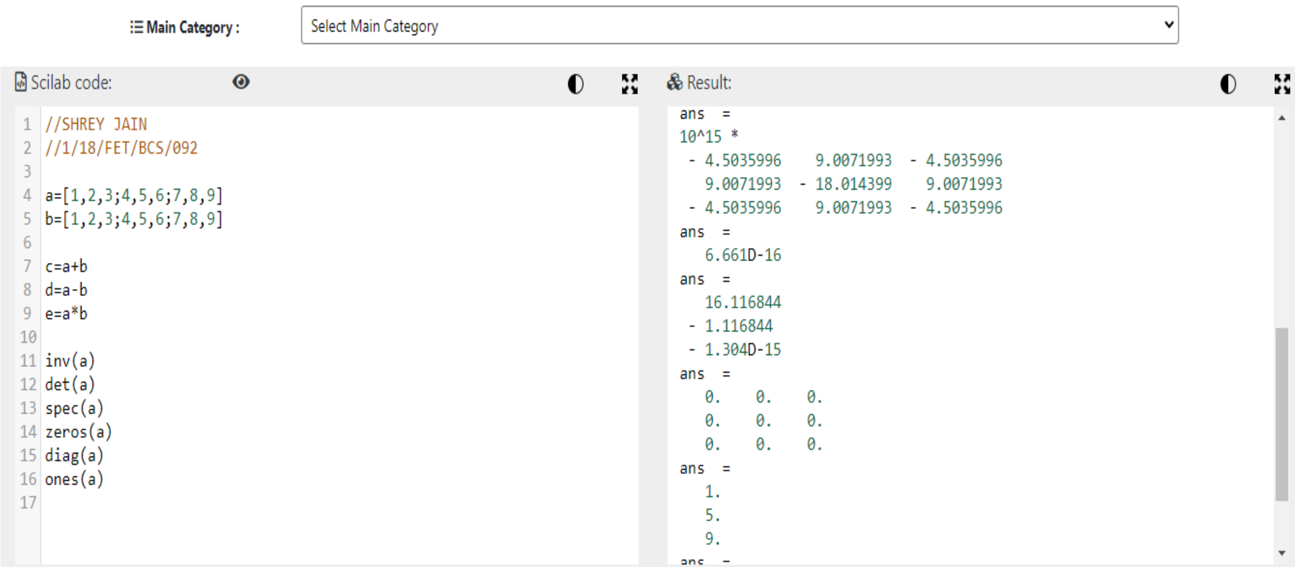
1.12 Zeros of matrix ----❼zeros(a)

1.13 Diagonal of matrix

----❼diag(a)

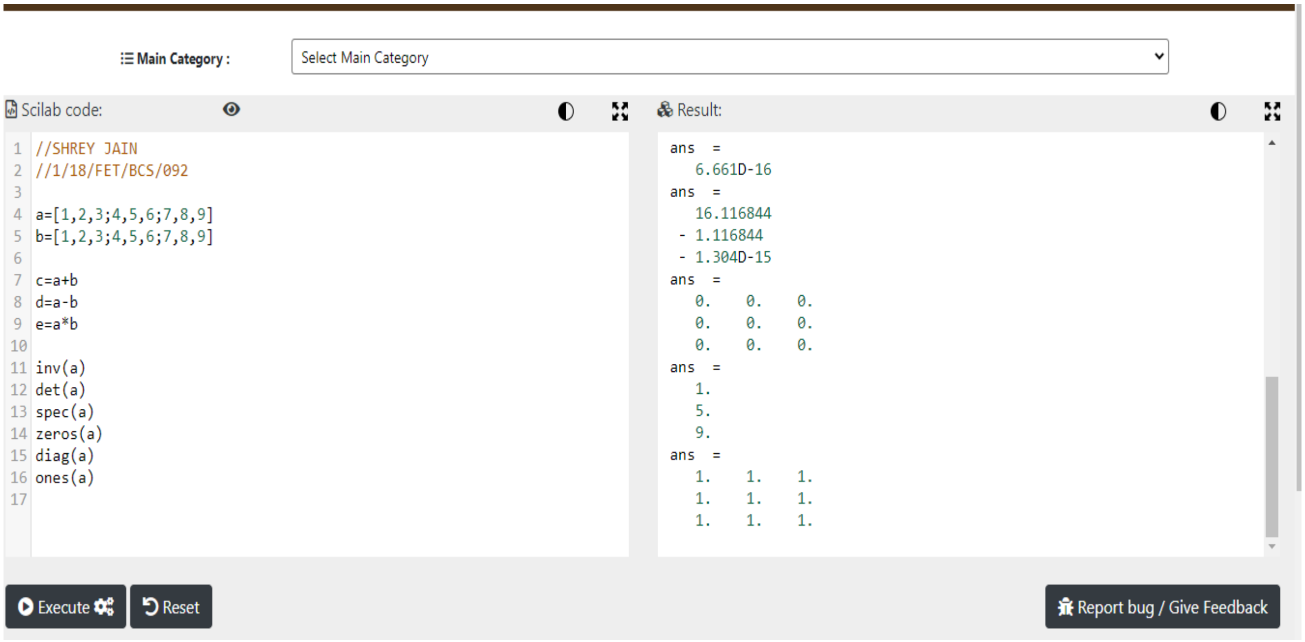
1.14 Ones of matrix

----❼ones(a)



```
//SHREY JAIN
//1/18/FET/BCS/092

a=[1,2,3;4,5,6;7,8,9]
b=[1,2,3;4,5,6;7,8,9]

c=a+b
d=a-b
e=a*b

inv(a)
det(a)
spec(a)
zeros(a)
diag(a)
ones(a)

```

```
ans  =
10^15 *
 - 4.5035996    9.0071993  - 4.5035996
   9.0071993  - 18.014399    9.0071993
 - 4.5035996    9.0071993  - 4.5035996
ans  =
   6.661D-16
ans  =
   16.116844
 - 1.116844
 - 1.304D-15
ans  =
   0.    0.    0.
   0.    0.    0.
   0.    0.    0.
ans  =
   1.
   5.
   9.
```



```
//SHREY JAIN
//1/18/FET/BCS/092

a=[1,2,3;4,5,6;7,8,9]
b=[1,2,3;4,5,6;7,8,9]

c=a+b
d=a-b
e=a*b

inv(a)
det(a)
spec(a)
zeros(a)
diag(a)
ones(a)

```

```
ans  =
   6.661D-16
ans  =
   16.116844
 - 1.116844
 - 1.304D-15
ans  =
   0.    0.    0.
   0.    0.    0.
   0.    0.    0.
ans  =
   1.
   5.
   9.
ans  =
   1.    1.    1.
   1.    1.    1.
   1.    1.    1.
```

EXPERIMENT: 2(b)

Aim: Find 30 terms using recursive function using Sci lab.

Recursive function:-

A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result at the end of each iteration. The process of recursive calls always has to end up in a call that is solved directly, without the need of invoking the function again. This step will always be needed in order to avoid a never ending loop.

---❼ u(1)=1

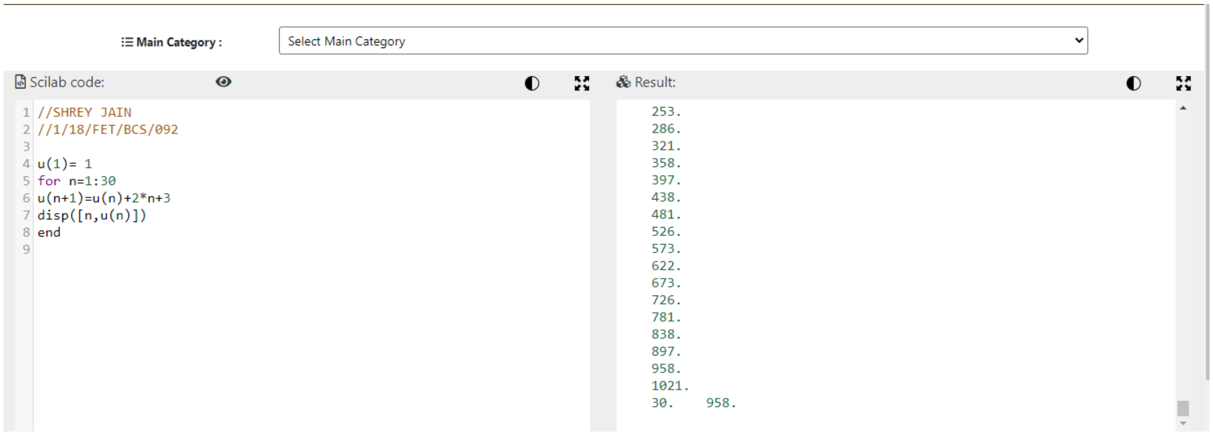
❼ for n=1:30

❼ u(n+1)=u(n)+2*n+3

❼ disp([n,u(n)])

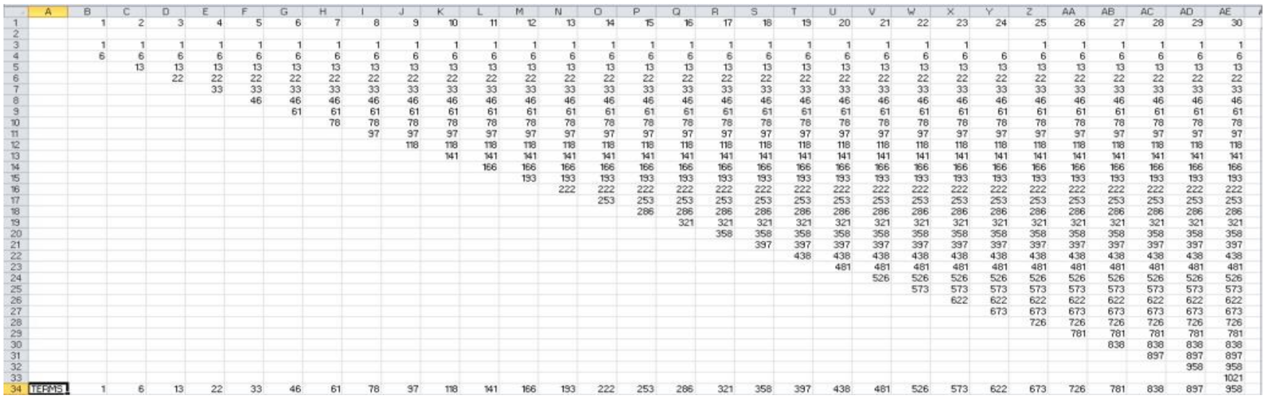
❼ end

Sci lab output:



```
//SHREY JAIN
//1/18/FET/BCS/092

u(1)= 1
for n=1:30
u(n+1)=u(n)+2*n+3
disp([n,u(n)])
end
```

```
253.
286.
321.
358.
397.
438.
481.
526.
573.
622.
673.
726.
781.
838.
897.
958.
1021.
30.    958.
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2																															
3		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1	1	1	1	1	1
4		6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5			13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
6				22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
7					33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33	33
8						46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46
9							61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61
10								78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78
11									97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97
12										118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118
13											141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141
14												166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166
15													193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193
16														222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222
17															253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253
18																286	286	286	286	286	286	286	286	286	286	286	286	286	286	286	286
19																	321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
20																		358	358	358	358	358	358	358	358	358	358	358	358	358	358
21																			397	397	397	397	397	397	397	397	397	397	397	397	397
22																				438	438	438	438	438	438	438	438	438	438	438	438
23																					481	481	481	481	481	481	481	481	481	481	481
24																						526	526	526	526	526	526	526	526	526	526
25																							573	573	573	573	573	573	573	573	573
26																								622	622	622	622	622	622	622	622
27																									673	673	673	673	673	673	673
28																										726	726	726	726	726	726
29																											781	781	781	781	781
30																												838	838	838	838
31																													897	897	897
32																														958	958
33																															1021
34	TERMS	1	6	13	22	33	46	61	78	97	118	141	166	193	222	253	286	321	358	397	438	481	526	573	622	673	726	781	838	897	958

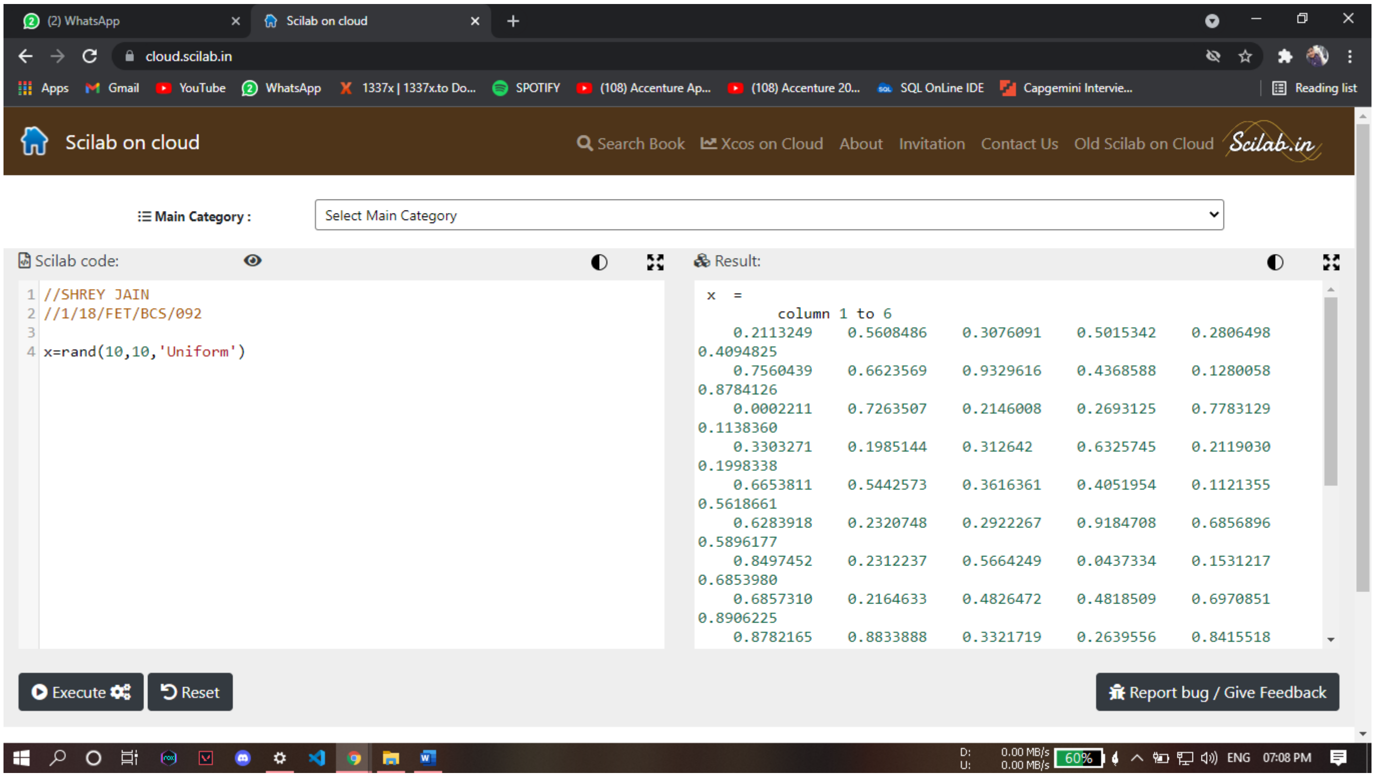
Experiment: 3

Aim: Find the computer generated random numbers using Sci lab.

Random numbers:-

A random number is a number chosen as if by chance from some specified distribution such that selection of a large set of these numbers reproduces the underlying distribution. Almost always, such numbers are also required to be independent, so that there are no correlations between successive numbers. Computer-generated random numbers are sometimes called pseudorandom numbers while the term "random" is reserved for the output of unpredictable physical processes. When used without qualification, the word "random" usually means "random with a uniform distribution" Other distributions are of course possible. For example, the Box-Muller transformation allows pairs of uniform random numbers to be transformed to corresponding random numbers having a two-dimensional normal distribution.

- - - ❼ X = rand(10,10,'Uniform')



```
//SHREY JAIN
//1/18/FET/BCS/092

x=rand(10,10,'Uniform')
```

```
 x  =
         column 1 to 6
    0.2113249    0.5608486    0.3076091    0.5015342    0.2806498
0.4094825
    0.7560439    0.6623569    0.9329616    0.4368588    0.1280058
0.8784126
    0.0002211    0.7263507    0.2146008    0.2693125    0.7783129
0.1138360
    0.3303271    0.1985144    0.312642     0.6325745    0.2119030
0.1998338
    0.6653811    0.5442573    0.3616361    0.4051954    0.1121355
0.5618661
    0.6283918    0.2320748    0.2922267    0.9184708    0.6856896
0.5896177
    0.8497452    0.2312237    0.5664249    0.0437334    0.1531217
0.6853980
    0.6857310    0.2164633    0.4826472    0.4818509    0.6970851
0.8906225
    0.8782165    0.8833888    0.3321719    0.2639556    0.8415518
```

The current random generator is set to a uniform random generator. Random numbers are uniformly distributed in the interval (0,1).

---❼Rand("normal")

The current random generator is set to a Gaussian (with mean 0 and variance 1) random number generator.

---❼Rand(‘info’)

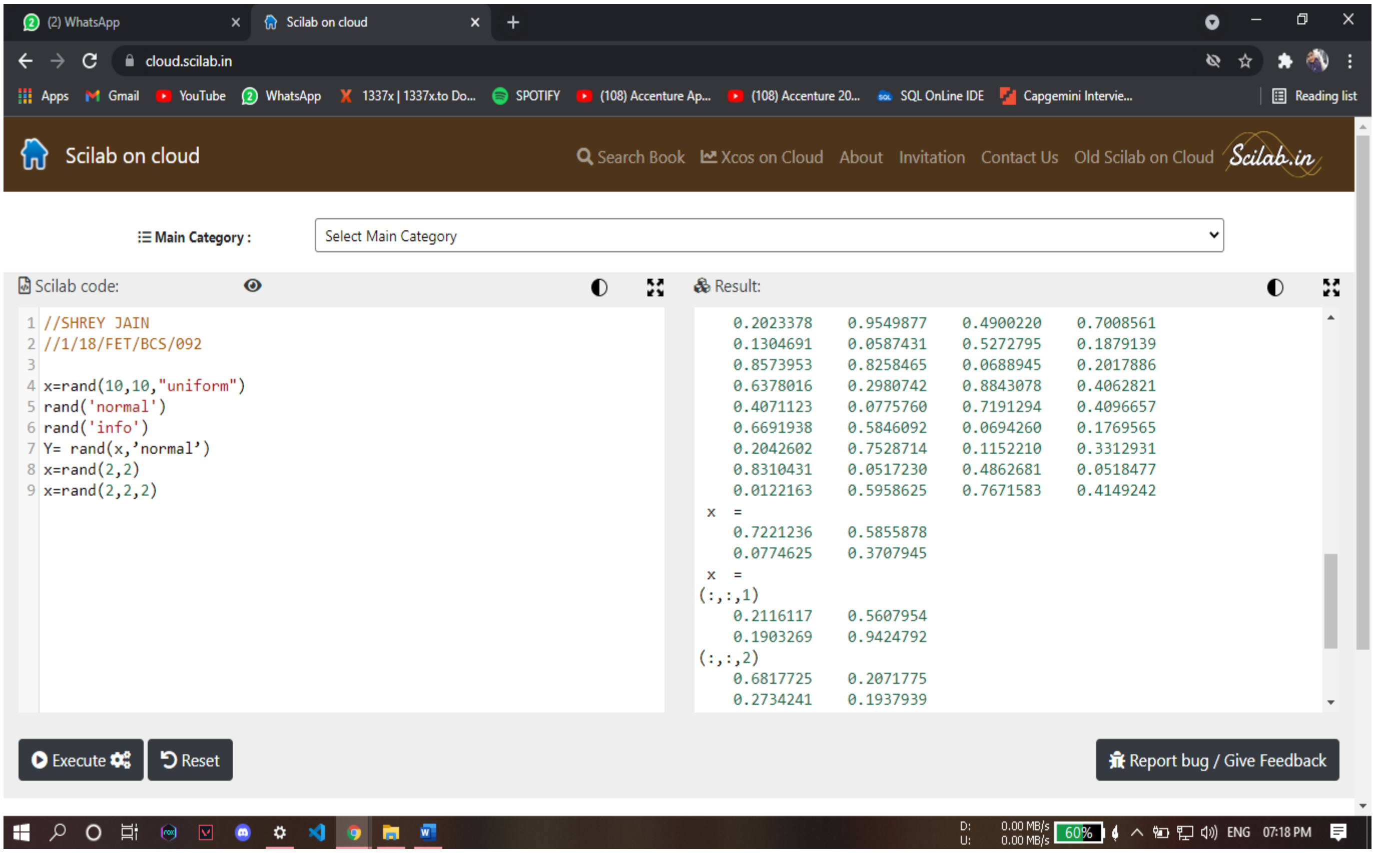
return the type of the default random generator ('uniform' or 'normal')

- - - ❼ Y = rand(x,’normal’)

returns the current value of the seed.

---❼X=rand(2,2)

---❼X=rand(2,2,2)



```
//SHREY JAIN
//1/18/FET/BCS/092

x=rand(10,10,"uniform")
rand('normal')
rand('info')
Y= rand(x,’normal’)
x=rand(2,2)
x=rand(2,2,2)
```

```
   0.2023378   0.9549877   0.4900220   0.7008561
   0.1304691   0.0587431   0.5272795   0.1879139
   0.8573953   0.8258465   0.0688945   0.2017886
   0.6378016   0.2980742   0.8843078   0.4062821
   0.4071123   0.0775760   0.7191294   0.4096657
   0.6691938   0.5846092   0.0694260   0.1769565
   0.2042602   0.7528714   0.1152210   0.3312931
   0.8310431   0.0517230   0.4862681   0.0518477
   0.0122163   0.5958625   0.7671583   0.4149242
 x  =
   0.7221236   0.5855878
   0.0774625   0.3707945
 x  =
(:,:,1)
   0.2116117   0.5607954
   0.1903269   0.9424792
(:,:,2)
   0.6817725   0.2071775
   0.2734241   0.1937939
```

Experiment: 4

Aim: Write a Sci- lab program for inventory control management.

Consider the following conditions:

There are 3-day lag between order and delivery/arrival. The merchandise is ordered at the end of the day(i) and received at the start of fourth day(i+3).

- For each unit of inventory the carrying cost of each night is Rs.0.75.
- Each unit out of stock when ordered results into the loss of goodwill worth Rs.2.0 per unit plus loss of Rs.16.00 net income, that would have resulted in its sale. Or a total loss of Rs. 18.00 per unit forever.
- The demand in a day can be for any number of units between 0 and 99, each equiprobable.
- There is never more than one replenishment order outstanding.
- Initially we have 115 units on hand and no reorder outstanding. What is

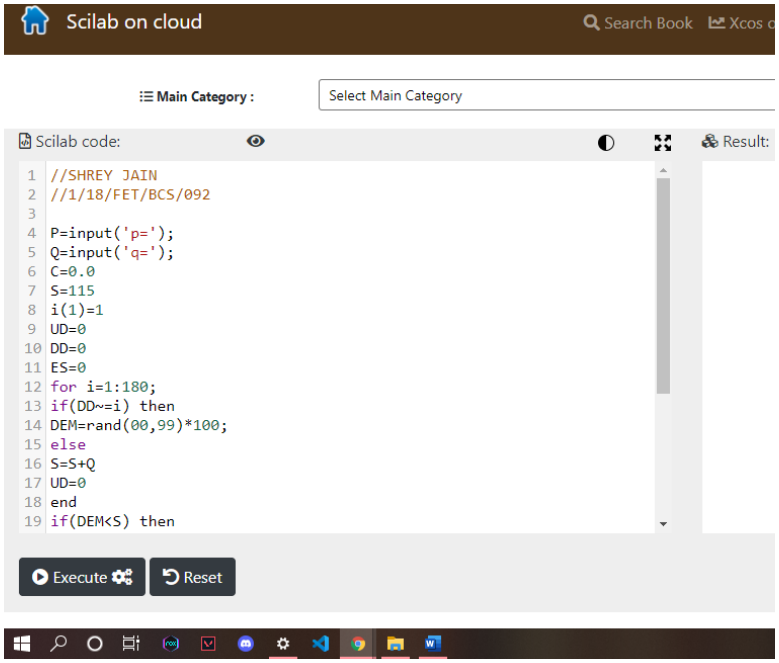
inventory condition?

- INVENTORY CONTROL : Webster's has defined Inventory as “The quantity of goods or the materials on hand”
- Goods or the materials are the essential element of any of the organization right from hospital, industry, private enterprise or the government department.
- Thus inventory control is the method of maintaining of stock at a level at which purchasing and stocking costs are at the lowest possible without interference with the supply.
- Thus it plays the vital role in maintaining the balance between the two. If the items like drugs are purchased in the large quantity, the supply can be made easily and immediately.
- The risk of the Out-Of-Stock is avoided.

```
P=input('p=');
Q=input('q=');
C=0.0
```

```
S=115
i(1)=1
UD=0
DD=0
ES=0
for i=1:180;
if(DD~=i) then
DEM=rand(00,99)*100;
else
S=S+Q
UD=0
end
if(DEM<S) then
S=S-DEM
C=C+(S)*75
else
C+((DEM)-(S))*18.0
S=0
UD=Q
DD=i+3
C=C+75.0
end
disp(C)
end
```

Sci lab output:



```
//SHREY JAIN
//1/18/FET/BCS/092

P=input('p=');
Q=input('q=');
C=0.0
S=115
i(1)=1
UD=0
DD=0
ES=0
for i=1:180;
if(DD~=i) then
DEM=rand(00,99)*100;
else
S=S+Q
UD=0
end
if(DEM<S) then
```

Experiment: 5

Aim: Given the age of different persons with their frequencies, calculate simple mean of age and plot graph between age and frequency.

Mean:-

The mean is the average of the numbers. It is easy to calculate: add up all the numbers, then divide by how many numbers there are. In other words it is the sum divided by the count.

Frequency:-

A frequency distribution is a list, table or graph that displays the frequency of various outcomes in a sample. Each entry in the table contains the frequency or count of the occurrences of values within a particular group or interval.

Graph:-

A simple graph usually shows the relationship between two numbers or measurements in the form of a grid. If this is a rectangular graph using Cartesian coordinate system, the two measurements will be arranged into two different lines at right angle to one another. One of these lines will be going up (the vertical axis). The other one will be going right (the horizontal axis). These lines (or axes, the plural of axis) meet at their ends in the lower left corner of the graph.

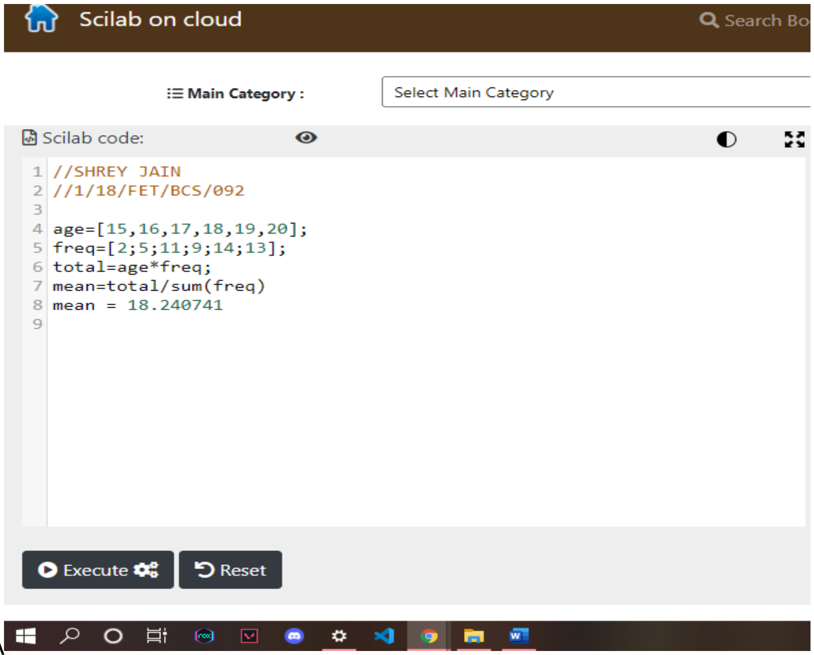
Both of these axes have tick marks along their lengths. You can think of each axis as a ruler drawn on paper. So each measurement is indicated by the length of the associated tick mark along the particular axis.

A graph is a kind of chart or diagram. However, a chart or a diagram may not relate one quantity to other quantities. Flowcharts and tree diagrams are charts or diagrams that are not graphs.

```
-->age=[15,16,17,18,19,20];

--> freq=[2;5;11;9;14;13];

--> total=age*freq;

--> mean=total/sum(freq)

 mean =
```

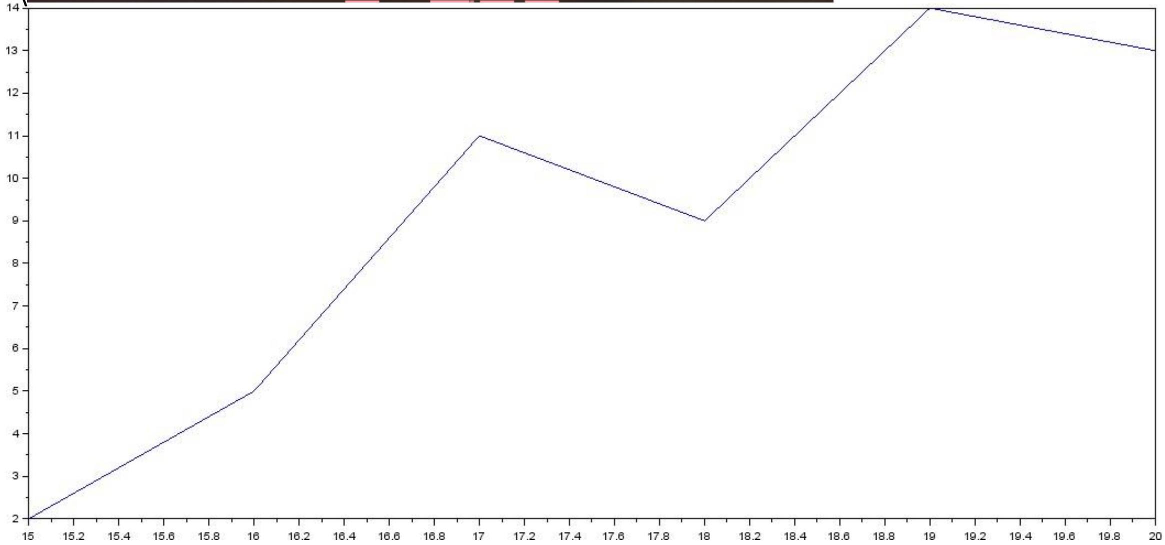
18.240741

--> Plot (age,freq)



```
//SHREY JAIN
//1/18/FET/BCS/092

age=[15,16,17,18,19,20];
freq=[2;5;11;9;14;13];
total=age*freq;
mean=total/sum(freq)
mean = 18.240741

```

\



Experiment: 6

<u>Aim: Find the expected profit if we have given 3 different types of profits and their probabilities.</u>

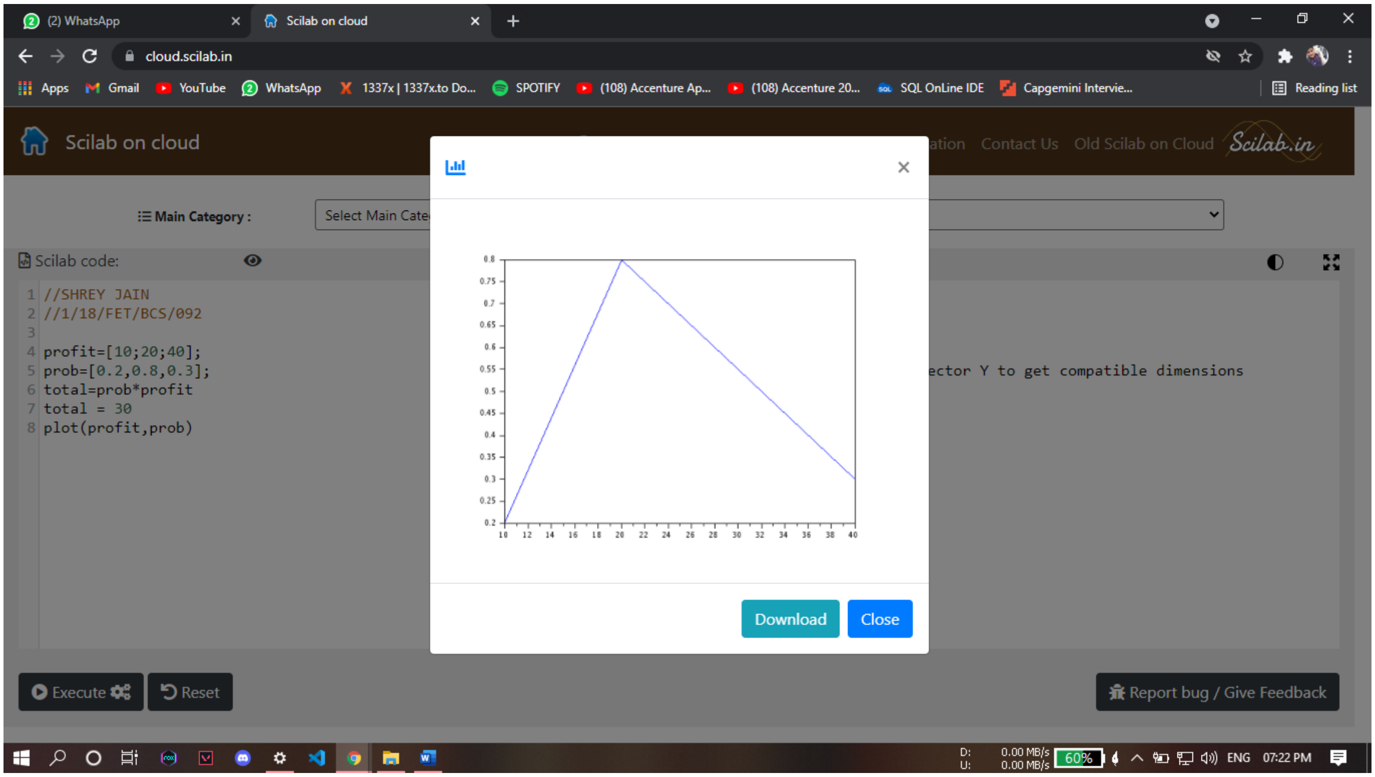
Profits 1= 10;	**prob 1=0.2**
Profits 2= 20;	**prob 2=0.8**
Profits 3= 40;	**prob 3=0.3**

Probability:-

Probability is a numerical description of how likely an event is to occur or how likely it is that a proposition is true. Probability is a number between 0 and 1, where, roughly speaking, 0 indicates impossibility and 1 indicates certainty. The higher the probability of an event, the more likely it is that the event will occur. A simple example is the tossing of a fair (unbiased) coin. Since the coin is fair, the two outcomes ("heads" and "tails") are both equally probable; the probability of "heads" equals the probability of "tails"; and since no other outcomes are possible, the probability of either "heads" or "tails" is 1/2 (which could also be written as 0.5 or 50%).

```
--> profit=[10;20;40];

--> prob=[0.2,0.8,0.3];

--> total=prob*profit
 total = 30.

--> plot(profit,prob)

WARNING: Transposing row vector Y to get compatible dimensions.
```

Sci lab outputs-



```
//SHREY JAIN
//1/18/FET/BCS/092

profit=[10;20;40];
prob=[0.2,0.8,0.3];
total=prob*profit
total = 30
plot(profit,prob)
```

```
ector Y to get compatible dimensions
```

Experiment: 7

Aim: To develop a program that finds out whether a tank will overflow or not, write the shape of the tannk, its dimensions and rate of flow.

If –else condition:-

The if/else statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.

The if/else statement is a part of JavaScript's "Conditional" Statements, which are used to perform different actions based on different conditions.

In JavaScript we have the following conditional statements:

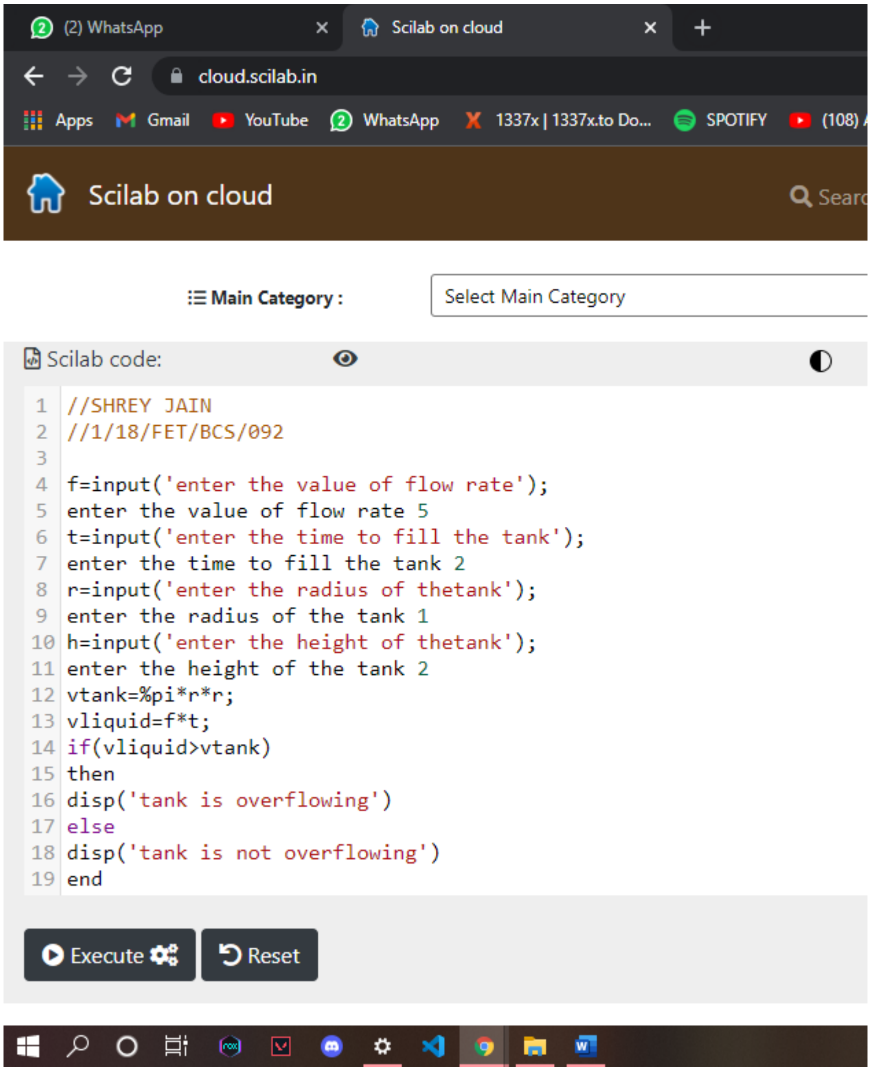
- Use if to specify a block of code to be executed, if a specified condition is true
- Use else to specify a block of code to be executed, if the same condition is false
- Use else if to specify a new condition to test, if the first condition is false
- Use switch to select one of many blocks of code to be executed

```
--> f=input('enter the value of flow rate');

enter the value of flow rate 5

--> t=input('enter the time to fill the tank');

enter the time to fill the tank 2

--> r=input('enter the radius of the tank');

enter the radius of the tank 1

--> h=input('enter the height of the tank'); enter the

height of the tank 2

--> vtank=%pi*r*r;

--> vliquid=f*t;

--> if(vliquid>vtank)

> then

> disp('tank is overflowing')

> else

> disp('tank is not overflowing')
```

> end

tank is overflowing

Sci lab outputs:



```
//SHREY JAIN
//1/18/FET/BCS/092

f=input('enter the value of flow rate');
enter the value of flow rate 5
t=input('enter the time to fill the tank');
enter the time to fill the tank 2
r=input('enter the radius of thetank');
enter the radius of the tank 1
h=input('enter the height of thetank');
enter the height of the tank 2
vtank=%pi*r*r;
vliquid=f*t;
if(vliquid>vtank)
then
disp('tank is overflowing')
else
disp('tank is not overflowing')
end
```

PRACTICAL: 8

Aim: Find Mean and Variance of rolling the disc for 6 times

- Mean:-

The mean is the average of the numbers. It is easy to calculate: add up all the numbers, then divide by how many numbers there are. In other words it is the sum divided by the count.

- Variance:-

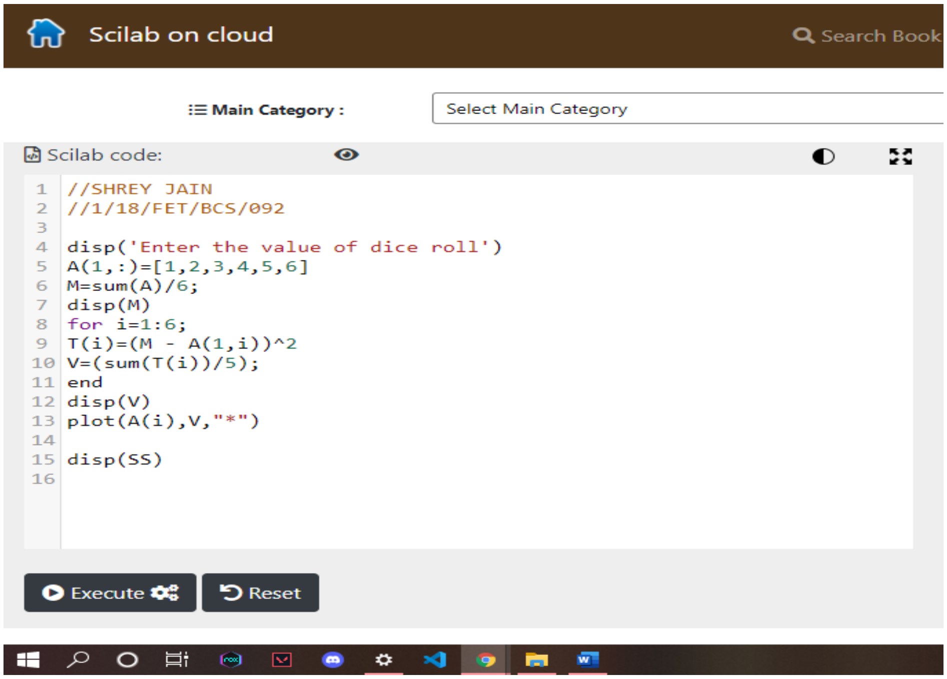
Variance (σ^2) in statistics is a measurement of the spread between numbers in a data set. That is, it measures how far each number in the set is from the mean and therefore from every other number in the set.

- How to calculate variance:-

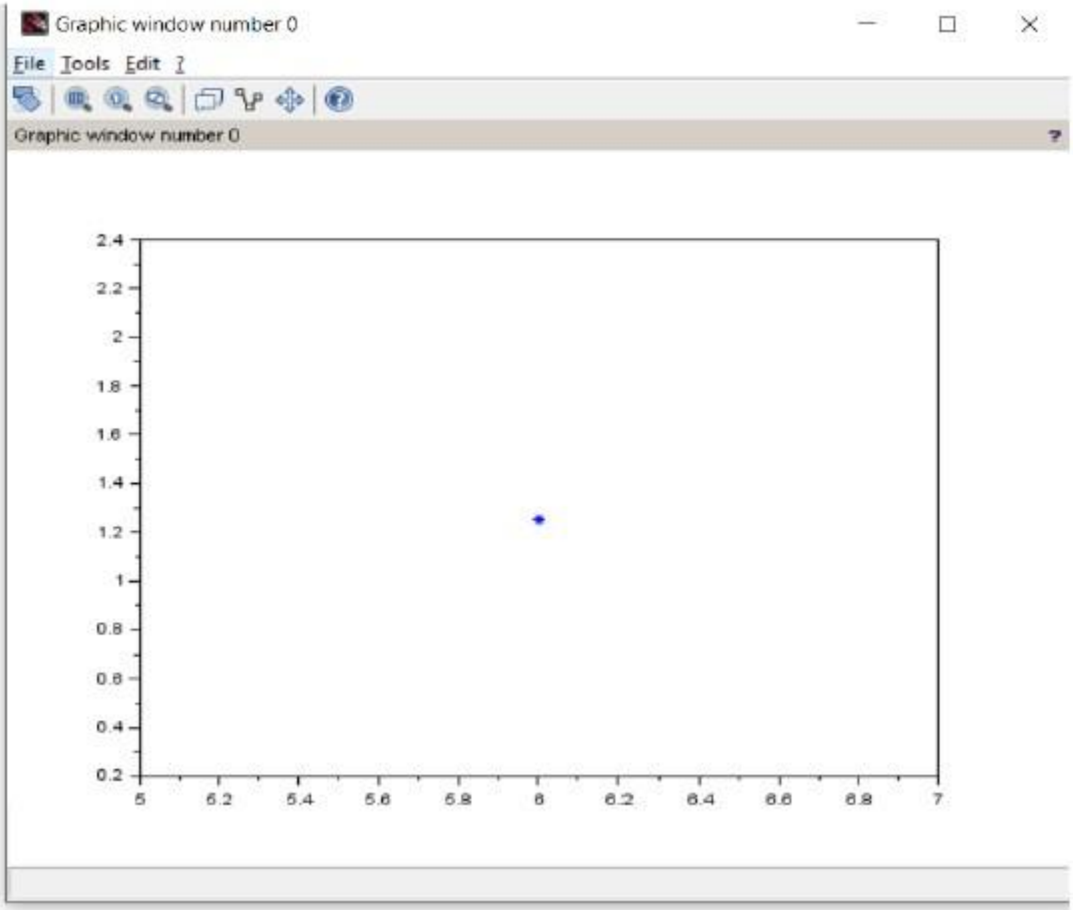
To calculate the variance follow these steps:

- Work out the mean (the simple average of the numbers)
- Then for each number: subtract the Mean and square the result (the squared difference).
- Then work out the average of those squared differences.

```
❼disp('Enter the value of dice roll')

❼A(1,:)=[1,2,3,4,5,6]

❼M=sum(A)/6;

❼disp(M)

❼ for i=1:6;

❼T(i)=(M - A(1,i))^2

❼V=(sum(T(i))/5);

❼ end

❼disp(V)

❼plot(A(i),V,"*")
```



```
//SHREY JAIN
//1/18/FET/BCS/092

disp('Enter the value of dice roll')
A(1,:)=[1,2,3,4,5,6]
M=sum(A)/6;
disp(M)
for i=1:6;
T(i)=(M - A(1,i))^2
V=(sum(T(i))/5);
end
disp(V)
plot(A(i),V,"*")

disp(SS)
```



Experiment: 9

Aim: To find the mean and variance for where given data where age is represented by Z and frequency by Y.

- Mean:-

The mean is the average of the numbers. It is easy to calculate: add up all the numbers, then divide by how many numbers there are. In other words it is the sum divided by the count.

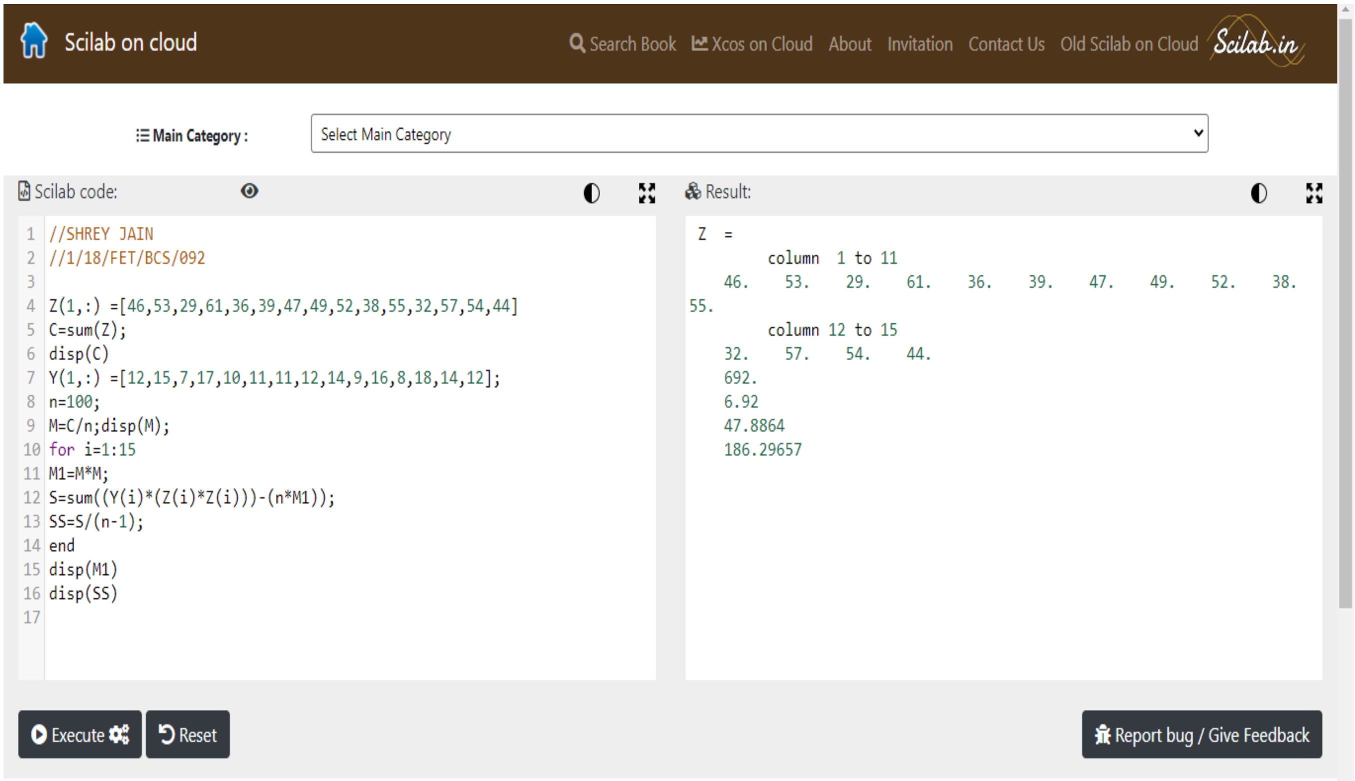
- Variance:-

Variance (σ^2) in statistics is a measurement of the spread between numbers in a data set. That is, it measures how far each number in the set is from the mean and therefore from every other number in the set.

- How is variance related to mean?

The variance is the average of the squared differences from the mean. To figure out the variance, first calculate the difference between each point and the mean; then, square and average the results. For example, if a group of numbers ranges from 1 to 10, it will have a mean of 5.5.

```
❼ Z(1,:) =[46,53,29,61,36,39,47,49,52,38,55,32,57,54,44]

❼ C=sum(Z);

❼ disp(C)

❼ Y(1,:) =[12,15,7,17,10,11,11,12,14,9,16,8,18,14,12];

❼ n=100;

❼ M=C/n;

❼ disp(M);

❼ for i=1:15

❼ M1=M*M;

❼S=sum((Y(i)*(Z(i)*Z(i)))-(n*M1));

❼ SS=S/(n-1);

❼ end

❼ disp(M1)

❼ disp(SS)
```



```
//SHREY JAIN
//1/18/FET/BCS/092

Z(1,:) =[46,53,29,61,36,39,47,49,52,38,55,32,57,54,44]
C=sum(Z);
disp(C)
Y(1,:) =[12,15,7,17,10,11,11,12,14,9,16,8,18,14,12];
n=100;
M=C/n;disp(M);
for i=1:15
M1=M*M;
S=sum((Y(i)*(Z(i)*Z(i)))-(n*M1));
SS=S/(n-1);
end
disp(M1)
disp(SS)
```

```
 Z  =
         column  1 to 11
    46.    53.    29.    61.    36.    39.    47.    49.    52.    38.
55.
         column 12 to 15
    32.    57.    54.    44.
    692.
    6.92
    47.8864
    186.29657
```

Experiment: 10(A)

Aim:To find the Covariance and Correlation.

- Correlation:-

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect. People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Nonetheless, the average weight of people 5'5'' is less than the average weight of people 5'6'', and their average weight is less than that of people 5'7'', etc. Correlation can tell you just how much of the variation in peoples' weights is related to their heights.

- Covariance:-

Covariance measures the directional relationship between the returns on two assets. A positive covariance means that asset returns move together while a negative covariance means they move inversely. Covariance is calculated by analysing at-return surprises (standard deviation from the expected return) or by multiplying the correlation between the two variables by the standard deviation of each variable.

- How to calculate covariance and correlation?

Population Covariance Formula

$$Cov(x,y) = \frac{\Sigma(x_i-\bar{x})(y_i-\bar{y})}{N}$$

Sample Covariance

$$Cov(x,y) = \frac{\Sigma(x_i-\bar{x})(y_i-y)}{N-1}$$

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

❼ A = [

4.0 2.0 0.60

4.2 2.1 0.59

3.9 2.0 0.58

4.3 2.1 0.62

4.1 2.2 0.63];

❼ S = [

0.025 0.0075 0.00175

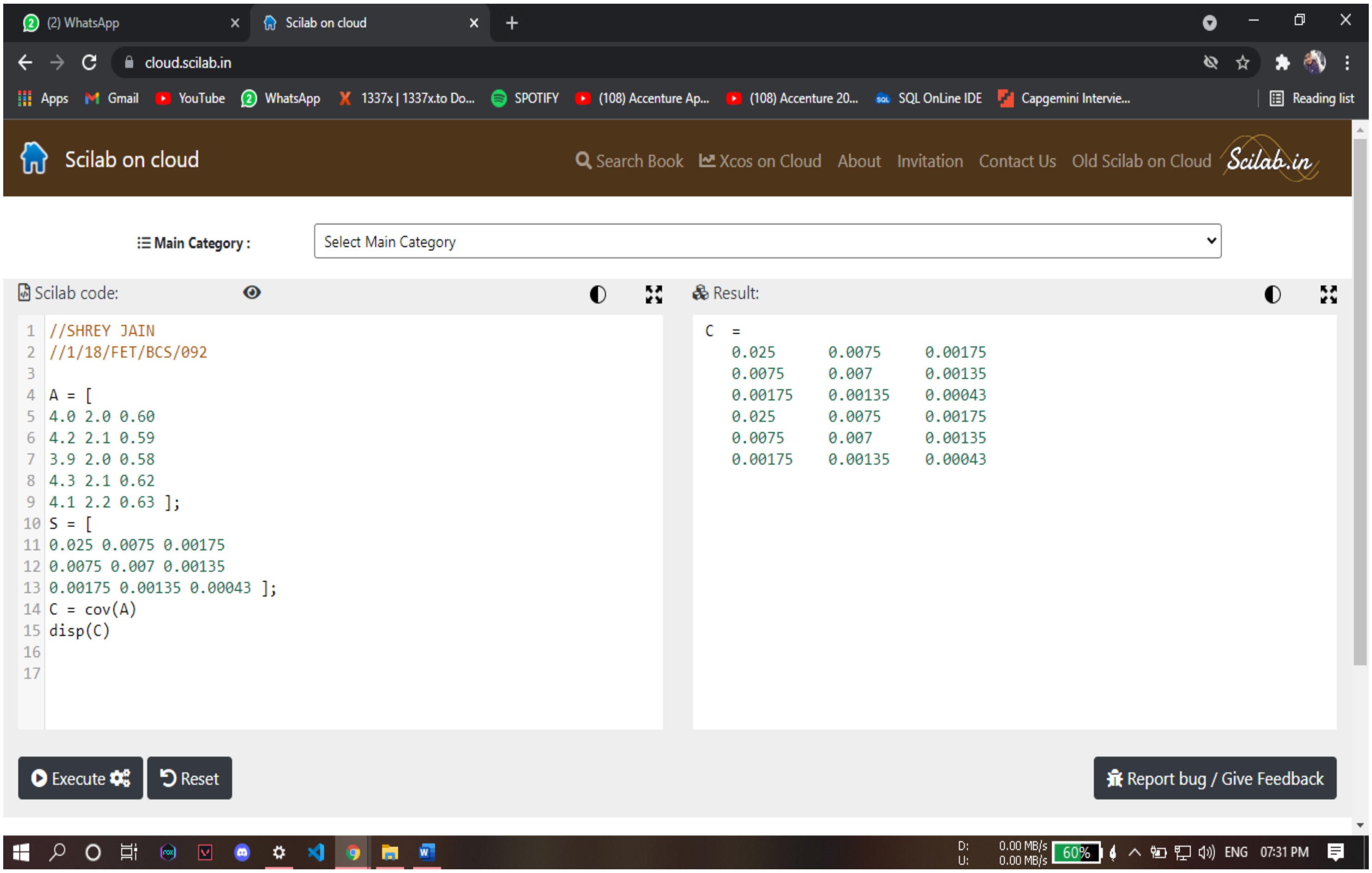
0.0075 0.007 0.00135

0.00175 0.00135 0.00043];

❼ C = cov(A)

❼ disp(C)

Sci lab output



```
//SHREY JAIN
//1/18/FET/BCS/092

A = [
4.0 2.0 0.60
4.2 2.1 0.59
3.9 2.0 0.58
4.3 2.1 0.62
4.1 2.2 0.63 ];
S = [
0.025 0.0075 0.00175
0.0075 0.007 0.00135
0.00175 0.00135 0.00043 ];
C = cov(A)
disp(C)
```

```
C  =
   0.025     0.0075    0.00175
   0.0075    0.007     0.00135
   0.00175   0.00135   0.00043
   0.025     0.0075    0.00175
   0.0075    0.007     0.00135
   0.00175   0.00135   0.00043
```

Experiment: 10(B)

Aim:To find the Covariance.

- **Covariance?**

Covariance measures the directional relationship between the returns on two assets. A positive covariance means that asset returns move together while a negative covariance means they move inversely. Covariance is calculated by analysing at-return surprises (standard deviation from the expected return) or by multiplying the correlation between the two variables by the standard deviation of each variable.

❼ x = [230; 181; 165; 150; 97; 192; 181; 189; 172; 170];

❼ y = [125; 99; 97; 115; 120; 100; 80; 90; 95; 125];

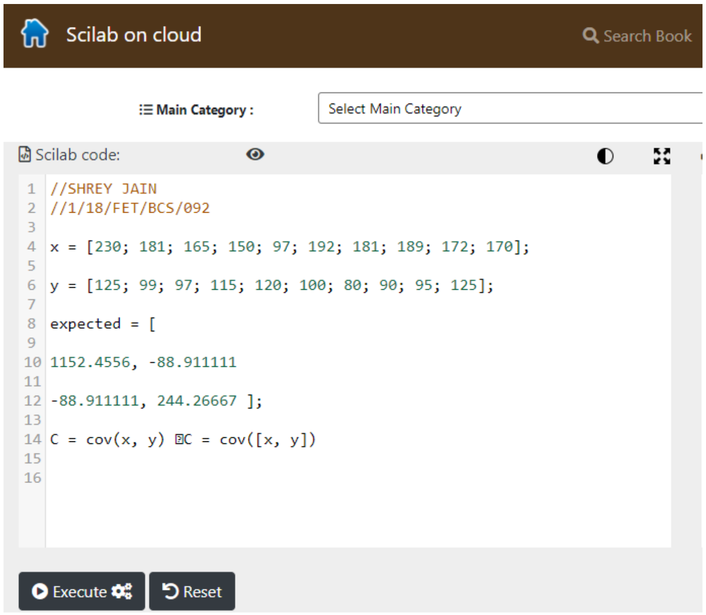
❼ expected = [

1152.4556, -88.911111

-88.911111, 244.26667];

❼ C = cov(x, y)
❼ C = cov([x, y])

Sci lab output:-



```
//SHREY JAIN
//1/18/FET/BCS/092

x = [230; 181; 165; 150; 97; 192; 181; 189; 172; 170];

y = [125; 99; 97; 115; 120; 100; 80; 90; 95; 125];

expected = [

1152.4556, -88.911111

-88.911111, 244.26667 ];

C = cov(x, y) □C = cov([x, y])


```

Experiment: 10(c)

Aim: To find the Correlation

- **Correlation:-**

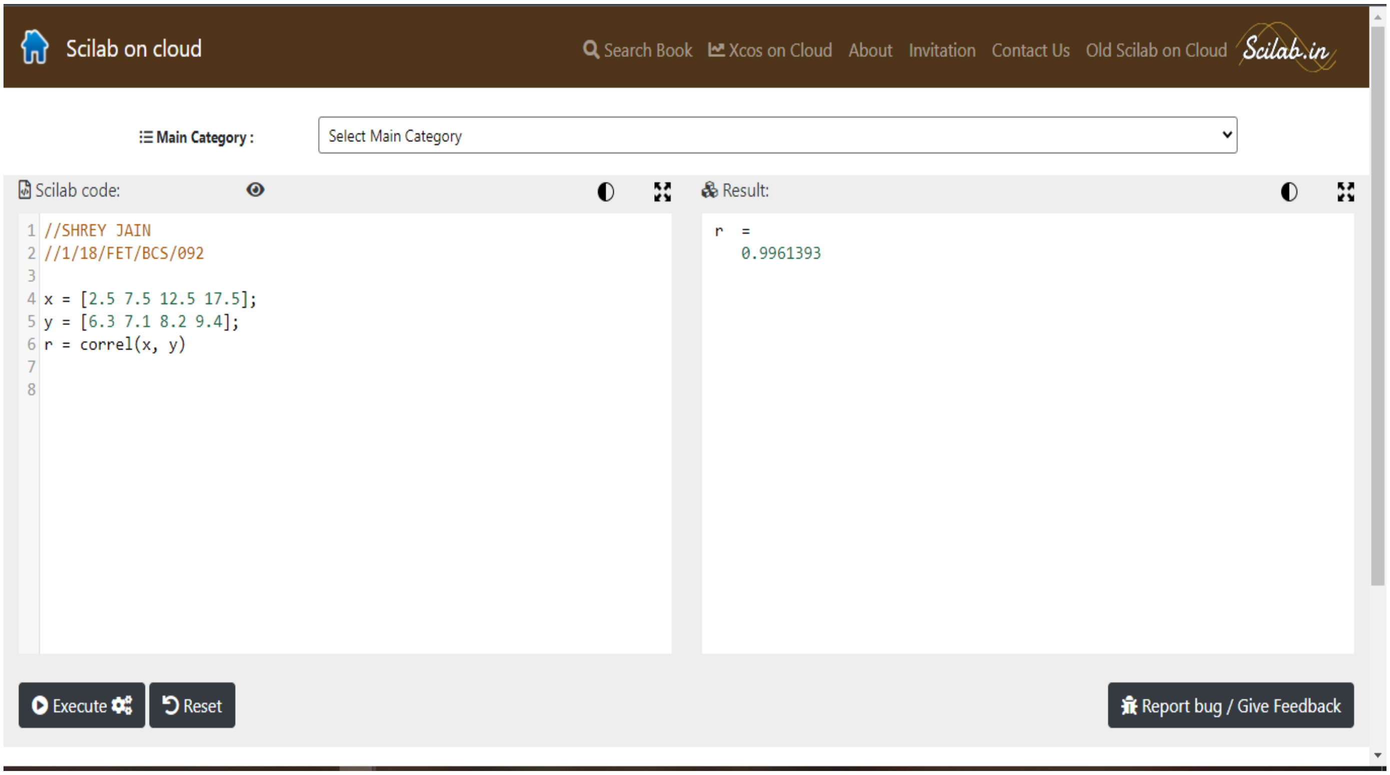
Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect. People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Nonetheless, the average weight of people 5'5'' is less than the average weight of people 5'6'', and their average weight is less than that of people 5'7'', etc. Correlation can tell you just how much of the variation in peoples' weights is related to their heights.

❼ x = [2.5 7.5 12.5 17.5];

❼ y = [6.3 7.1 8.2 9.4];

❼ r = correl(x, y)

Sci lab output-



```
//SHREY JAIN
//1/18/FET/BCS/092

x = [2.5 7.5 12.5 17.5];
y = [6.3 7.1 8.2 9.4];
r = correl(x, y)
```

```
r  =
   0.9961393
```

Experiment: 11

Aim: We have a about vehicle performance, Miles per gallon is represented By matrix m and corresponding weight of car is represented by W matrix. Find Covariance and Correlation between these parameters. Plot the data set.

- Correlation:-

Correlation is a statistical technique that can show whether and how strongly pairs of variables are related. For example, height and weight are related; taller people tend to be heavier than shorter people. The relationship isn't perfect. People of the same height vary in weight, and you can easily think of two people you know where the shorter one is heavier than the taller one. Nonetheless, the average weight of people 5'5'' is less than the average weight of people 5'6'', and their average weight is less than that of people 5'7'', etc. Correlation can tell you just how much of the variation in peoples' weights is related to their heights.

Although this correlation is fairly obvious your data may contain unsuspected correlations. You may also suspect there are correlations, but don't know which are the strongest. An intelligent correlation analysis can lead to a greater understanding of your data

- Covariance:-

Covariance measures the directional relationship between the returns on two assets. A positive covariance means that asset returns move together while a negative covariance means they move inversely. Covariance is calculated by analysing at-return surprises (standard deviation from the expected return) or by multiplying the correlation between the two variables by the standard deviation of each variable.

❼ m = [. . . .]

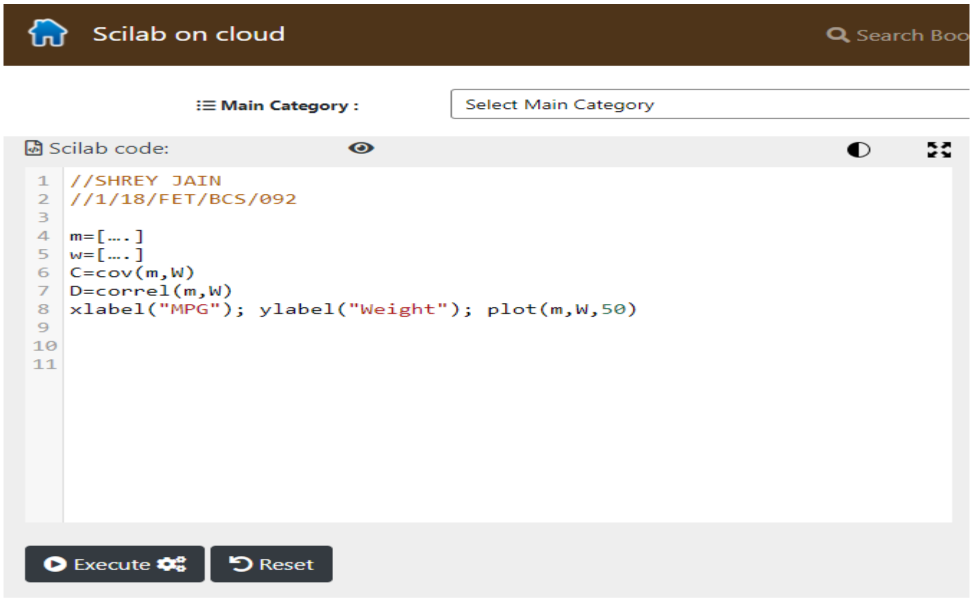
❼ w = [… .]

❼ C=cov(m,W)

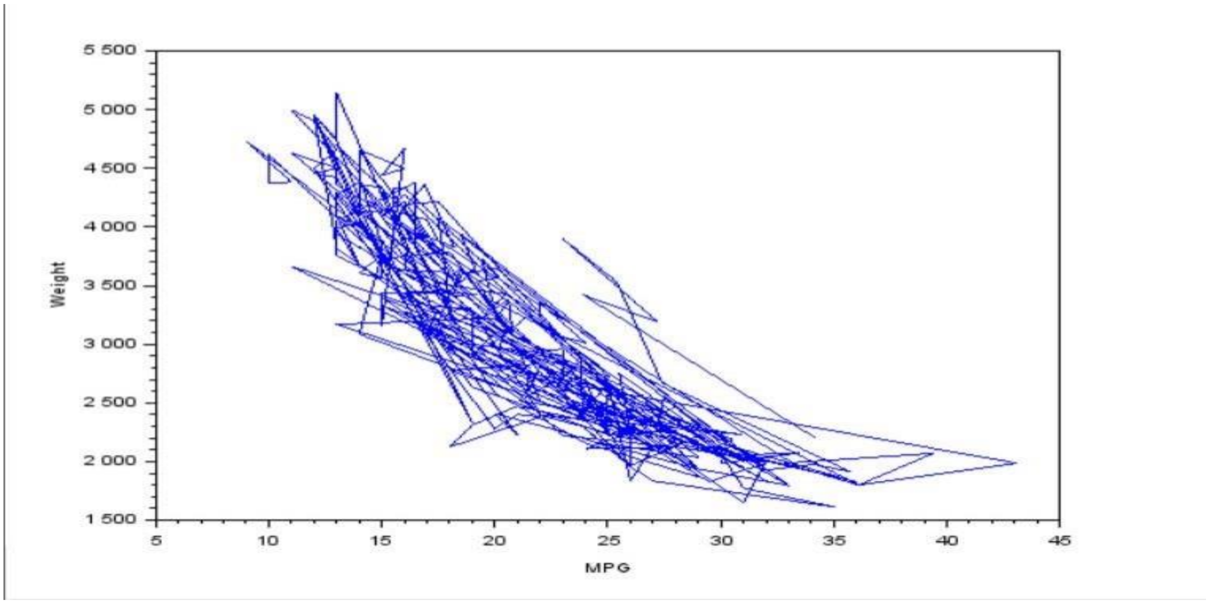
❼ D=correl(m,W)

❼ xlabel("MPG"); ylabel("Weight"); plot(m,W,50)

Sci lab output-



```
//SHREY JAIN
//1/18/FET/BCS/092

m=[….]
w=[….]
C=cov(m,W)
D=correl(m,W)
xlabel("MPG"); ylabel("Weight"); plot(m,W,50)
```



Experiment: 12

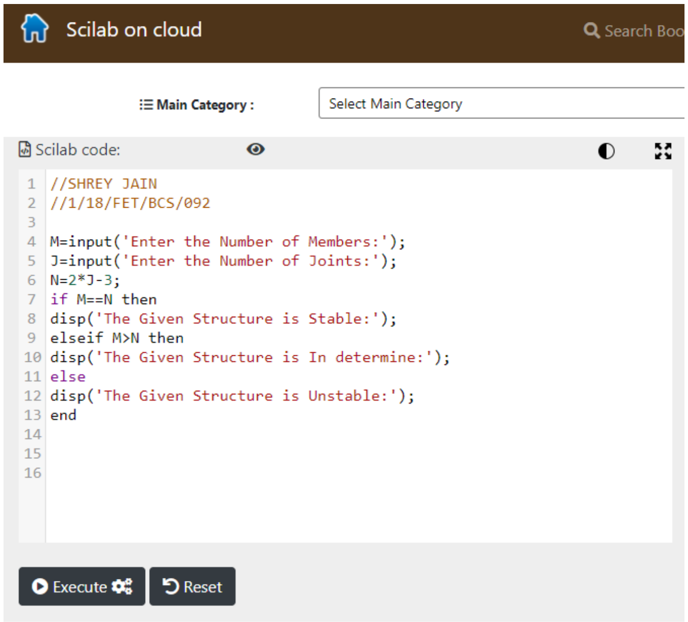
Aim:-Write a program to find the structural stability of the given truss bridge.

Simulation steps to check the stability of the bridge:
STEP 1: Assume any definite shape (shapes made of straight lines).

STEP 2: Stability of the truss shall be determined using the formula, m = 2j-3 Where ‘m’ – No. of members in the given structure (nos.) ‘j’ – No. of joints in the given structure (nos.)

STEP 3: Conditions m< 2j-3 Unstable [Deficient Truss] m = 2j-3 Stable or Statically determinate [Perfect Truss] m > 2j-3 Statically indeterminate [Redundant Truss]

```
❼M=input('Enter the Number of Members:');

❼J=input('Enter the Number of Joints:');

❼N=2*J-3;

❼ if M==N then

❼disp('The Given Structure is Stable:');

❼elseif M>N then

❼disp('The Given Structure is In determine:');

❼else

❼disp('The Given Structure is Unstable:');
```



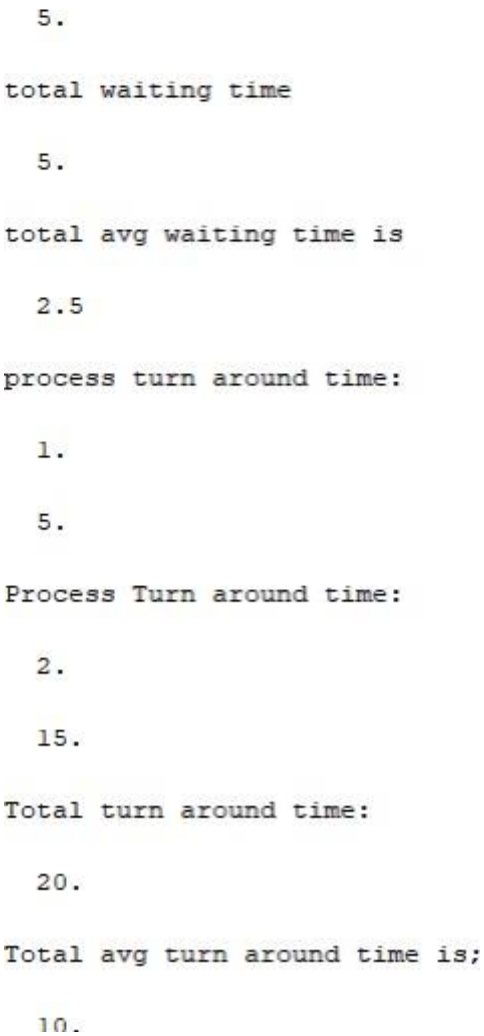
```
//SHREY JAIN
//1/18/FET/BCS/092

M=input('Enter the Number of Members:');
J=input('Enter the Number of Joints:');
N=2*J-3;
if M==N then
disp('The Given Structure is Stable:');
elseif M>N then
disp('The Given Structure is In determine:');
else
disp('The Given Structure is Unstable:');
end
```

Experiment- 13

Aim: Write a program to implement single server.

```
queue. function FCFS()
n=input("Enter the no. of process :")
disp(" enter the burst time of process :")
for i=1:n
disp(i,"Process"
) b(i)=input(" ")
a(i)=i
end
w(1)=0
avg=0
disp(w(1),a(1),"process waiting time:")
for i= 2:n
w(i)=b(i-1)+w(i-1)
disp(w(i),a(i),"Process waiting time")
avg=avg+w(i)
end
disp(avg,"total waiting time") disp(avg/n,"total avg waiting time is") tat(1)=b(1)
avg1=b(1)
disp(tat(1),a(1),"process turn around time:")
for k= 2:n
tat(k)=tat(k-1)+b(k)
disp(tat(k),a(k),"Process Turn around time:")
avg1=avg1+tat(k)
end
disp(avg1,"Total turn around time: ")
disp(avg1/n,"Total avg turn around time is; ")
//exec(‘C:\Users\Administrator\Desktop\mona sainiprjct\new
//prg\fcfs.sci', -1)
endfunction
FCFS()
```

```
Enter the no. of process :2


  enter the burst time of process :

 Process

   1.
 5


 Process

   2.
 10


 process waiting time:

   1.

   0.

 Process waiting time

   2.

   5.
```



```
  5.

total waiting time

  5.

total avg waiting time is

  2.5

process turn around time:

  1.

  5.

Process Turn around time:

  2.

  15.

Total turn around time:

  20.

Total avg turn around time is;

  10.
```

Experiment: 14

Aim:Write a program for pure pursuit problem using SCI lab.

Pure pursuit:-

Pure pursuit is a tracking algorithm that works by calculating the curvature that will move a vehicle from its current position to some goal position. The whole point of the algorithm is to choose a goal position that is some distance ahead of the vehicle on the path. The name pure pursuit comes from the analogy that we use to describe the method. We tend to think of the vehicle as chasing a point on the path some distance ahead of it - it is pursuing that moving point. That analogy is often used to compare this method to the way humans drive. We tend to look some distance in front of the car and head toward that spot. This look ahead distance changes as we drive to reflect the twist of the road and vision occlusions.

Simulating fighter aircraft hitting a bomber-

Station of a pure purse problem ample fighter aircraft sights an enemy bomber and flies directly toward it A in order to catch up with the bomber and destroy it The bomber the target) continuous lying (along a specified curve to the frontier the pure has to change its direction to keep pointed toward the target. We are interested in determining the attack course of the fighter and in mewing how long it would take for it to catch up with the bomber

If the target flies along a straight time, the problem can be solved directly with analytic techniques (The proof of such a closed-form expression which gives the course of the pure, when the target is in straight line is le as an exercise for you. Problem 1-2) However, in the path of the target is curved, the problem is much more difficult and normally cannot be solved directly. We will use simulation to solve this problem, under the following simplifying conditions

1.The target and the pursuer are flying in the same horizontal plane when the fighter first sights the bomber, and both stay in the plane. This makes the pursuit model two- dimensional.

2. The lighter's speed the target path (e. its position as a function of time is specified

Enter a fixed time span Ar (every minute, in this case the fighter changes its direction in order to printer toward the bomber.

Let us introduce a rectangular coordinate system coincident with the horizontal plane in which the two aircraft are tying We choose the post due south of the night and due west of the target at the beginning of the be givenpur) as the origin of the coordinate system.

Explanation-

C analytically we could not make a long-term prediction about the pat chat the fighter plane would take (in the initial position and path or are). But by simulation we were able to make the computer go there the inset-to predictions for as many instants as we wanted W possible only because we knew the basic process involved, namely at any particular instant.

The fighter plane system under study is essential for all simulation. Such knowledge of the simple strategy, of pune redirecting sed intervals of time, while the target goes on ou ma y himself toward the tar predetermined pa AE effort to evade the pursuer, is called pure pursuit. in many situations, the strategy wed by the pursuer is more sophisticated.

```
Sci code: Xb=[100,110,120,129,140,149,158,168,179,188,198,209,219,226,234,240]

Yb=[0,3,6,10,15,20,26,32,37,34,30,27,23,19,16,14]

Xf=[];

Yf=[];

Xf[1]=0;

Yf[1]=50;

S=20;

dist=0;

for i=1:15;

plot(xb(i),yb(i),’r’’);

title(‘pure persuit problem’);

plot(xf(i),yf(i),’g’’);

y=yb(i)-yf(i);

x=xb(i)-xf(i);

dist=sqrt(y^2+x^2);
```

if(dist<=12)

display('bomber destroyed at',s,i);

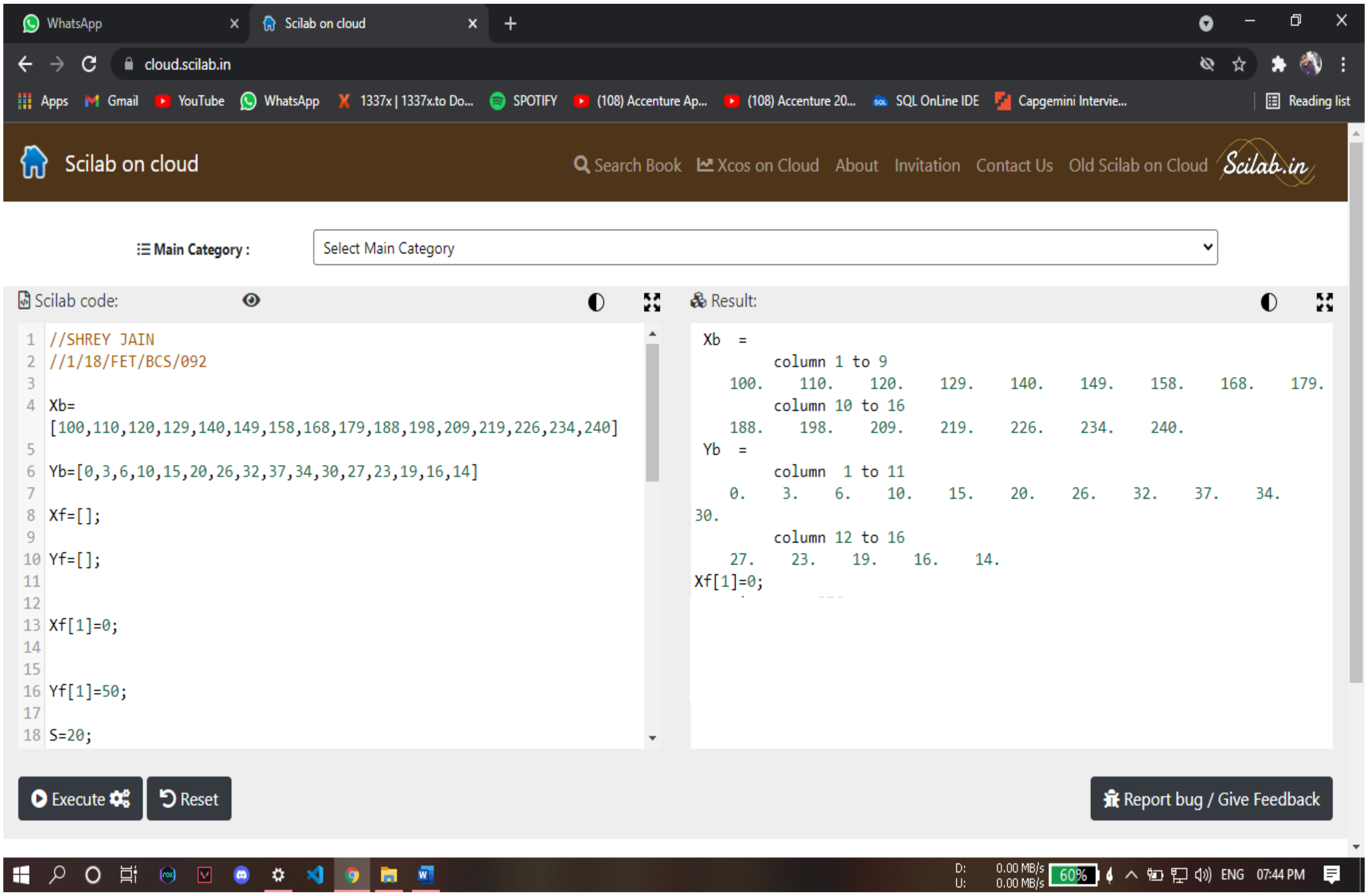
break;

end

xf(i+1)=xf(i)+s*((xb(i)-xf(i))/dist);

yf(i+1)=yf(i)+s*((yb(i)-yf(i))/dist); end

Sci lab output-



```
//SHREY JAIN
//1/18/FET/BCS/092

Xb=
[100,110,120,129,140,149,158,168,179,188,198,209,219,226,234,240]

Yb=[0,3,6,10,15,20,26,32,37,34,30,27,23,19,16,14]

Xf=[];

Yf=[];


Xf[1]=0;


Yf[1]=50;

S=20;
```

```
 Xb  =
         column 1 to 9
   100.    110.    120.    129.    140.    149.    158.    168.    179.
         column 10 to 16
   188.    198.    209.    219.    226.    234.    240.
 Yb  =
         column  1 to 11
   0.    3.    6.    10.    15.    20.    26.    32.    37.    34.
30.
         column 12 to 16
   27.    23.    19.    16.    14.
Xf[1]=0;
```