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: Anshul Bhardwaj Kumar

Roll No.: 1/18/FET/BCS/080

Branch: FET CSE (N)

Section: 7CSB

Submitted To:-

Faculty Name: Dr. Brijesh

Designation: Professor

Department: CSE

INDEX

Practical No.	Aim of Practical	Teacher's Signature
1	Introduction to Sci Lab.	
2(a) **2(b)**	Study of matrix operation using Sci lab. Find 30 terms using recursive function using Sci lab.	
3	Find the computer generated random numbers using Sci lab.	
4	Write a Sci- lab program for inventory control management.	
5	Given the age of different persons with their frequencies, calculate simple mean of age and plot graph between age and frequency.	
6	Find the expected profit if we have given 3 different types of profits and their probabilities.	
7	To develop a program that finds out whether a tank will overflow or not, write the shape of the tank, its dimensions and rate of flow.	
8	Find Mean and Variance of rolling the disc for 6 times.	
9	To find the mean and variance for where given data where age is represented by Z and frequency by Y.	
10	We have a about vehicle performance, Miles per gallon is represented By matrix m and corresponding weight of car is represented by W matrix.	

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 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 particular problems. The Scilab language allows to dynamically compile and link other languages such as Fortran and C: this way, external libraries can be used as if they were a part of Scilab built-in features. Scilab also interfaces LabVIEW, a platform and development environment for a visual programming language from National Instruments. From the license point of view, Scilab is a free software in the sense that the user does not pay for it and Scilab is an open source software, provided under the Cecill license [2]. The software is distributed with source code, so that the user has an access to Scilab most internal aspects. Most of the time, the user downloads and installs, a binary version of Scilab since the Scilab consortium provides Windows, Linux and Mac OS executable versions. An online help is provided in many local languages. From a scientific point of view, Scilab comes with many features. At the very beginning of Scilab, features were focused on linear algebra. But, rapidly, the number of features extended to cover many areas of scientific computing.

The following is a short list of its capabilities:

- Linear algebra, sparse matrices
- Polynomials and rational functions
- Interpolation approximation
- Linear, quadratic optimization
- Ordinary Differential Equation solver and Differential Algebraic Equations solver
- Classic and robust control, Linear Matrix Inequality optimization
- Differentiable and non-differentiable optimization,
- Signal processing
- Statistics
- Scilab provides many graphics features, including a set of plotting functions, which allow to create 2D and 3D plots as well as user interfaces. The Xcos environment provides an hybrid dynamic systems modeler and simulator.

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
- Null or Zero Matrix
- Transpose of a Matrix

1. Creation of matrix

1.1 Create a row scaler matrix

```
- - - - ❼ a =  [7,8,9]
```

1.2 Create a vector scaler matrix

```
- - - - ❼ b = [ 4 ;  6; 7]
```

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 Multiplication ----

❼ e = a * b

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

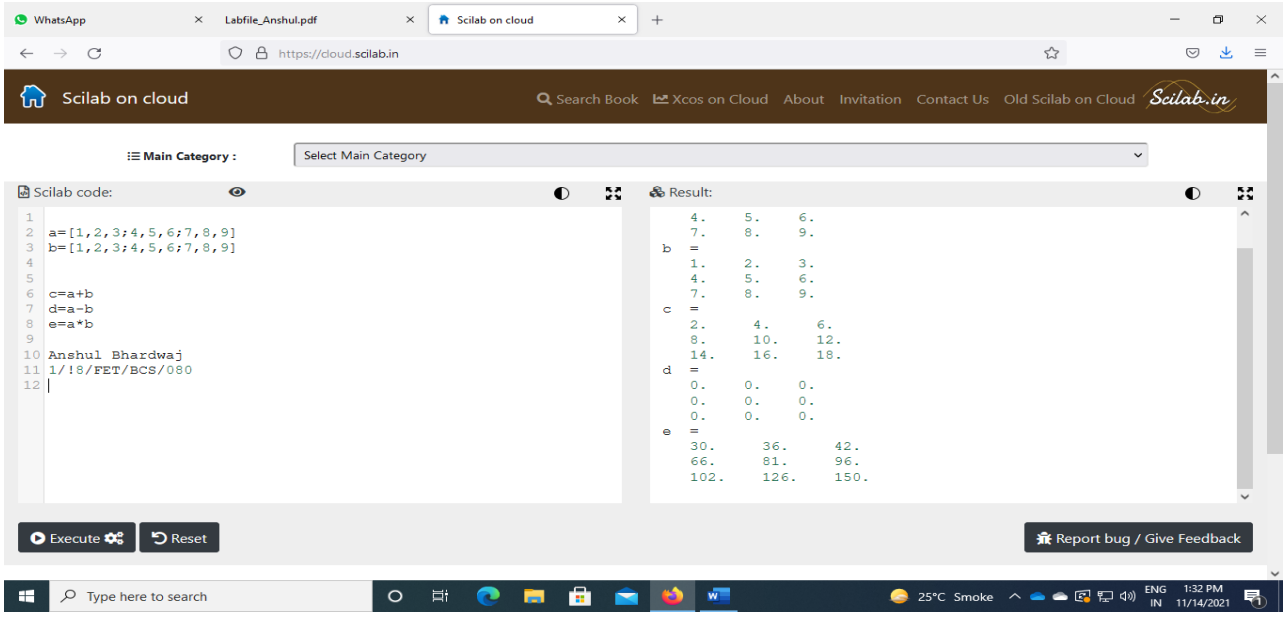
----❼det(a)

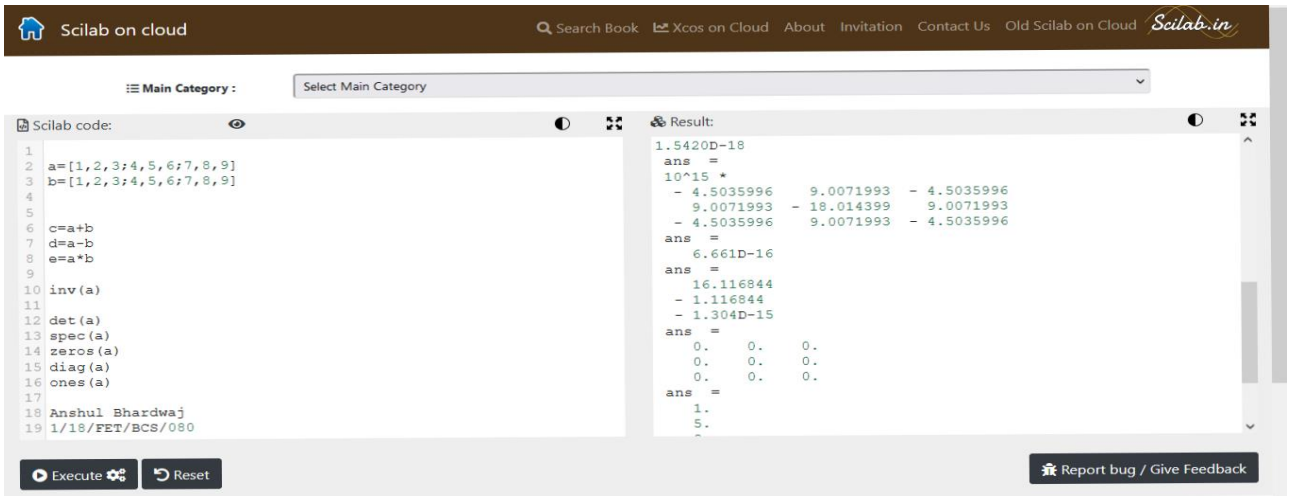
1.11 Eiyen

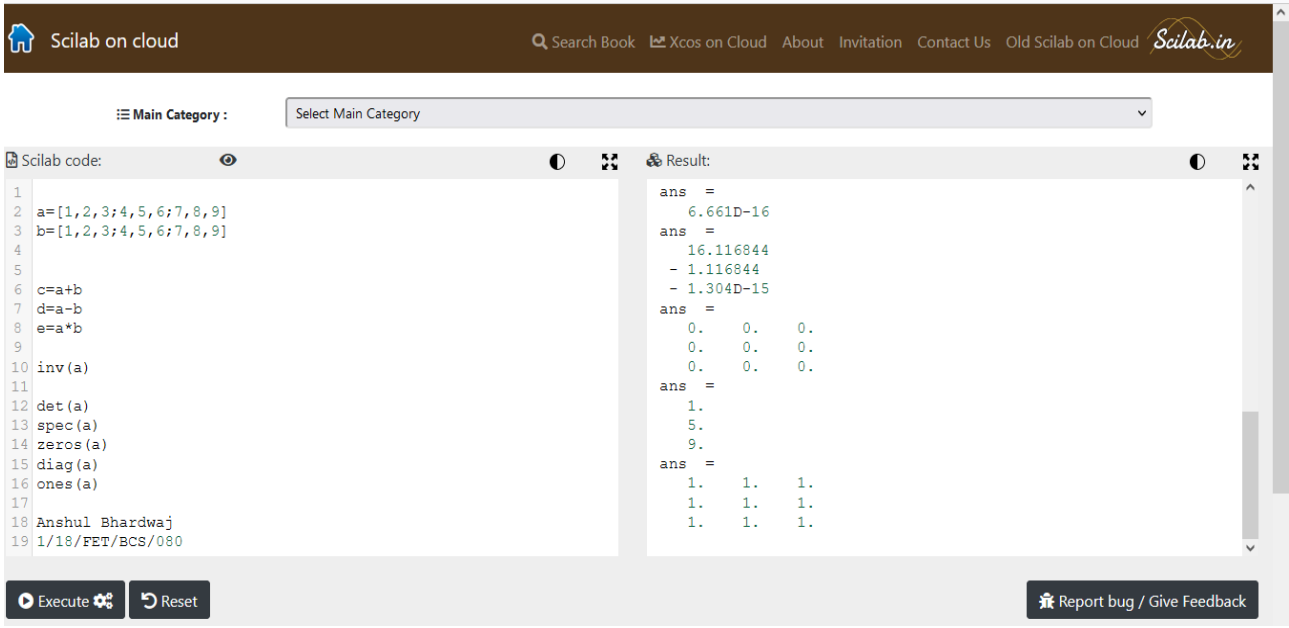
----❼spec(a)

1.12 Zeros of matrix

----❼zeros(a)




Scilab on cloud
Search Book
Xcos on Cloud
About
Invitation
Contact Us
Old Scilab on Cloud
Scilab.in
Main Category :
Select Main Category
Scilab code:
Result:
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)
Anshul Bhardwaj
1/18/FET/BCS/080
1.5420D-18
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.
Execute
Reset
Report bug / Give Feedback


Scilab on cloud
Search Book
Xcos on Cloud
About
Invitation
Contact Us
Old Scilab on Cloud
Scilab.in
Main Category :
Select Main Category
Scilab code:
Result:
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)
Anshul Bhardwaj
1/18/FET/BCS/080
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.
Execute
Reset
Report bug / Give Feedback

1.13 Diagonal of matrix -

---➐diag(a)

1.14 Ones of matrix

----➐ones(a)

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 f o r n=1:30

❼ u(n+1)=u(n)+2*n+3

❼ disp([n,u(n)])

❼ e n d

Main Category : Select Main Category

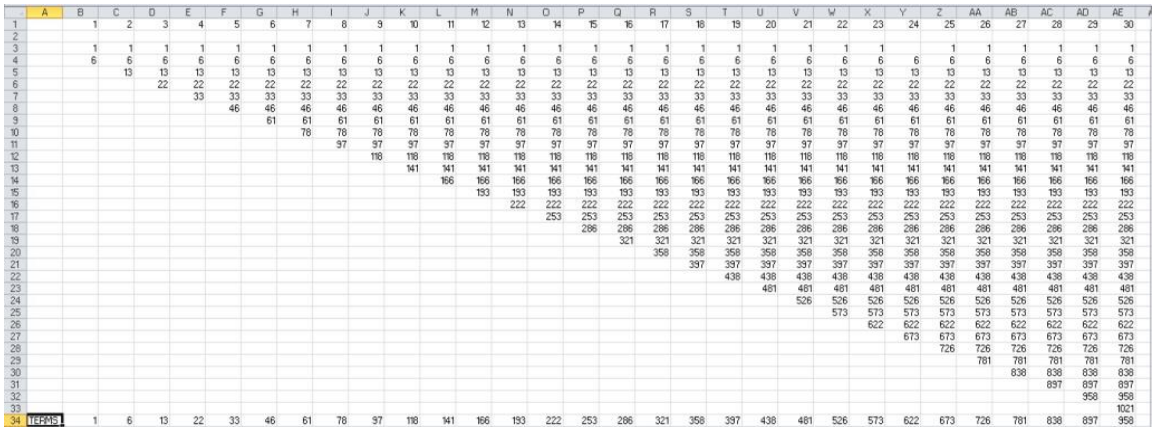
Scilab code:

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

Anshul Bhardwaj
1/18/FET/BCS/080
```

Result:

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

Sci lab output:



	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
	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
	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
		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
			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
				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
					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
						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
							78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78	78
								97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97	97
									118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118	118
										141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141	141
											166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166	166
												193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193	193
													222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222	222
														253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253	253
															286	286	286	286	286	286	286	286	286	286	286	286	286	286	286	286
																321	321	321	321	321	321	321	321	321	321	321	321	321	321	321
																	358	358	358	358	358	358	358	358	358	358	358	358	358	358
																		397	397	397	397	397	397	397	397	397	397	397	397	397
																			438	438	438	438	438	438	438	438	438	438	438	438
																				481	481	481	481	481	481	481	481	481	481	481
																					526	526	526	526	526	526	526	526	526	526
																						573	573	573	573	573	573	573	573	573
																							622	622	622	622	622	622	622	622
																								673	673	673	673	673	673	673
																									726	726	726	726	726	726
																										781	781	781	781	781
																											838	838	838	838
																												897	897	897
																													958	958
																														1021
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')

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

```
--> 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   0.113836
   0.3303271   0.1985144   0.312642    0.6325745   0.211903    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   0.685398
   0.685731    0.2164633   0.4826472   0.4818509   0.6970851   0.8906225
   0.8782165   0.8833888   0.3321719   0.2639556   0.8415518   0.5042213
   0.068374    0.6525135   0.5935095   0.4148104   0.4062025   0.3493615

        column 7 to 10

   0.3873779   0.537623    0.587872    0.6488563
   0.9222899   0.1199926   0.4829179   0.9923191
   0.9488184   0.2256303   0.2232865   0.050042
   0.3435337   0.6274093   0.8400886   0.7485507
   0.3760119   0.7608433   0.1205996   0.4104059
   0.7340941   0.0485566   0.2855364   0.6084526
   0.2615761   0.672395    0.8607515   0.8544211
   0.4993494   0.2017173   0.8494102   0.0642647
   0.2638578   0.3911574   0.5257061   0.8279083
   0.5253563   0.8300317   0.993121    0.9262344
```

---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')

```
--> rand("normal")

--> rand('info')
 ans  =

 normal
```

---❼ Y = rand(x,'normal')

returns the current value of the seed.

```
--> y=rand(x,'normal')
 y  =


         column 1 to 6

   1.7487359  -1.3770621   0.2301981  -0.8575198   2.4976095   1.1316248
   1.8651792   0.7042731  -2.7290777  -0.1043591  -1.2875914   0.3759656
   0.1645912  -0.9063738  -0.2563031   0.2973099   0.6450695  -1.3667445
  -1.035891    0.2634747  -0.5003797   0.5308516   0.6696589  -0.0346505
   0.9182207   1.2296215   1.1937458  -1.5404673  -0.4483985  -1.3850463
  -0.9355485  -1.1579022  -1.5206394  -0.3966362  -1.5316782   0.3828792
   0.0259118  -0.4577385   1.8655071   0.5163255  -0.7218988  -1.6280467
   0.2720404   0.0168437   0.1910551   0.0075659  -2.3544971  -0.4386207
   0.7953703  -0.5875092  -1.3189197   1.0422456  -0.5485232   0.7757721
  -1.681167   -1.4029475   0.9307226   2.6705108   0.0207588  -0.6024774
```

```
         column 7 to 10

  -1.1049895   0.0979236  -1.0179837   0.282304
  -1.7970057   1.1997935   0.7794314   0.5000272
   0.7604477   0.2946056  -1.9598191  -2.0049863
   1.0562604  -0.382814    0.3252542   1.2498744
  -0.65881     1.1474321   0.0927515   1.1758126
   1.2431698  -0.6155043  -0.5864591   1.0197259
   0.1068464   0.2685505   1.2311603  -0.257048
  -0.9032336  -0.2032709   0.3027452  -1.0665698
   0.7469882   0.878063    0.6697461  -0.1595813
  -1.2220922   0.4345772  -0.269772   -2.696727
```

---❼X=rand(2,2)

```
--> x=rand(2,2)
 x  =

   0.7131577  -1.2140244
  -0.4213568  -0.7860841
```

---❼X=rand(2,2,2)

```
--> x=rand(2,2,2)
 x  =

(:,:,1)

  -2.3432411  -2.0499751
  -0.6921925   0.4215026
(:,:,2)

  -0.4183469  -0.2768419
  -0.481796   -0.0518056
```

Sci lab output-

```
--> 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   0.113836
   0.3303271   0.1985144   0.312642    0.6325745   0.211903    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   0.685398
   0.685731    0.2164633   0.4826472   0.4818509   0.6970851   0.8906225
   0.8782165   0.8833888   0.3321719   0.2639556   0.8415518   0.5042213
   0.068374    0.6525135   0.5935095   0.4148104   0.4062025   0.3493615

         column 7 to 10

   0.3873779   0.537623    0.587872    0.6488563
   0.9222899   0.1199926   0.4829179   0.9923191
   0.9488184   0.2256303   0.2232865   0.050042
   0.3435337   0.6274093   0.8400886   0.7485507
   0.3760119   0.7608433   0.1205996   0.4104059
   0.7340941   0.0485566   0.2855364   0.6084526
   0.2615761   0.672395    0.8607515   0.8544211
   0.4993494   0.2017173   0.8494102   0.0642647
   0.2638578   0.3911574   0.5257061   0.8279083
   0.5253563   0.8300317   0.993121    0.9262344

--> rand("normal")
```

Scilab 6.0.2 Console

```
--> rand('info')
 ans  =

 normal


--> y=rand(x,'normal')
 y  =


         column 1 to 6

   1.7487359  -1.3770621   0.2301981  -0.8575198   2.4976095   1.1316248
   1.8651792   0.7042731  -2.7290777  -0.1043591  -1.2875914   0.3759656
   0.1645912  -0.9063738  -0.2563031   0.2973099   0.6450695  -1.3667445
  -1.035891    0.2634747  -0.5003797   0.5308516   0.6696589  -0.0346505
   0.9182207   1.2296215   1.1937458  -1.5404673  -0.4483985  -1.3850463
  -0.9355485  -1.1579022  -1.5206394  -0.3966362  -1.5316782   0.3828792
   0.0259118  -0.4577385   1.8655071   0.5163255  -0.7218988  -1.6280467
   0.2720404   0.0168437   0.1910551   0.0075659  -2.3544971  -0.4386207
   0.7953703  -0.5875092  -1.3189197   1.0422456  -0.5485232   0.7757721
  -1.681167   -1.4029475   0.9307226   2.6705108   0.0207588  -0.6024774
```

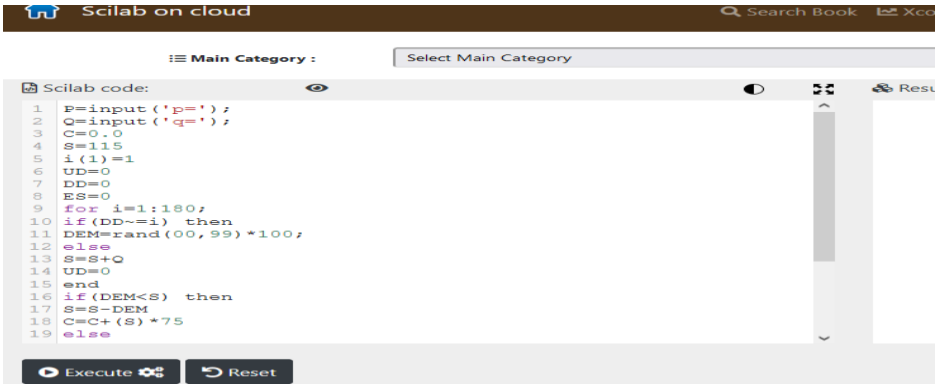
Scilab 6.0.2 Console

```
         column 7 to 10

  -1.1049895   0.0979236  -1.0179837   0.282304
  -1.7970057   1.1997935   0.7794314   0.5000272
   0.7604477   0.2946056  -1.9598191  -2.0049863
   1.0562604  -0.382814    0.3252542   1.2498744
  -0.65881     1.1474321   0.0927515   1.1758126
   1.2431698  -0.6155043  -0.5864591   1.0197259
   0.1068464   0.2685505   1.2311603  -0.257048
  -0.9032336  -0.2032709   0.3027452  -1.0665698
   0.7469882   0.878063    0.6697461  -0.1595813
  -1.2220922   0.4345772  -0.269772   -2.696727


--> x=rand(2,2)
 x  =

   0.7131577  -1.2140244
  -0.4213568  -0.7860841


--> x=rand(2,2,2)
 x  =

(:,:,1)

  -2.3432411  -2.0499751
  -0.6921925   0.4215026
(:,:,2)

  -0.4183469  -0.2768419
  -0.481796   -0.0518056
```

Experiment: 4

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

```
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:

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)

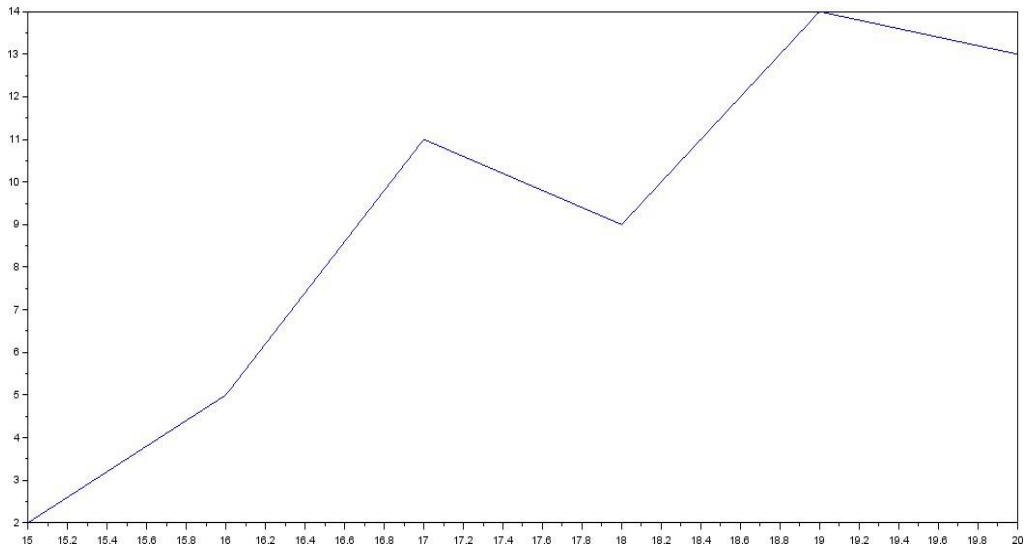
Scilab on cloud

Search Boo

Main Category : Select Main Category

Scilab code:

```
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)

Anshul Bhardwaj
1/18/FET/BCS/080
```



Experiment: 6

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

Profits 1= 10; **prob 1=0.2**

Profits 2= 20; **prob 2=0.8**

Profits 3= 40; **prob 3=0.3**

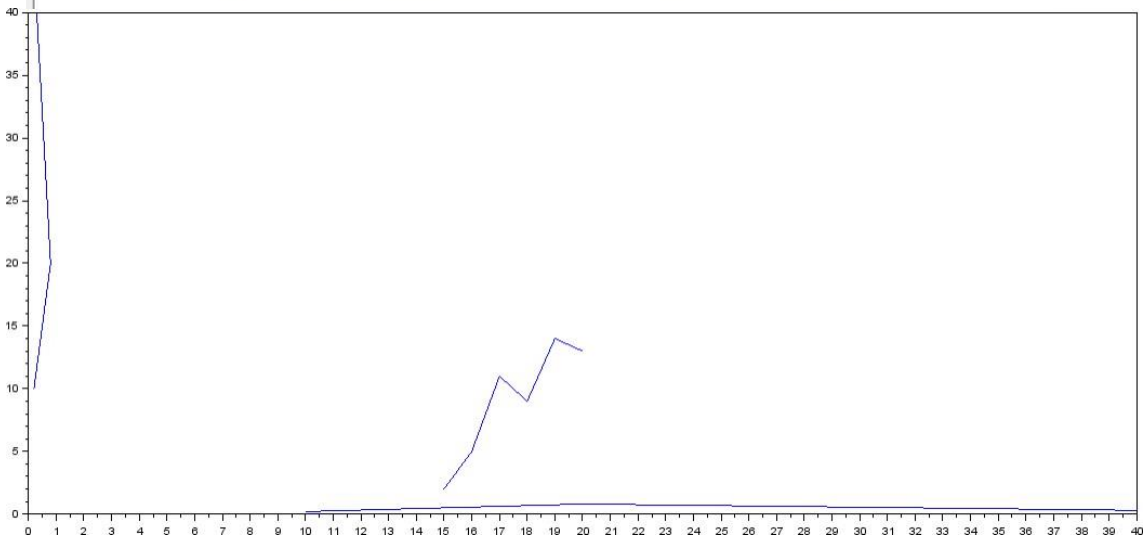
Probability:-

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-

```
Scilab 6.0.2 Console

--> 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
```



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.

```
--> 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

Output

tank is overflowing

Sci lab outputs:

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.

```
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,"*")
```

s 8.sce (C:\Users\Richa\Desktop\s 8.sce) - SciNotes

File Edit Format Options Window Execute ?

s 8.sce (C:\Users\Richa\Desktop\s 8.sce) - SciNotes

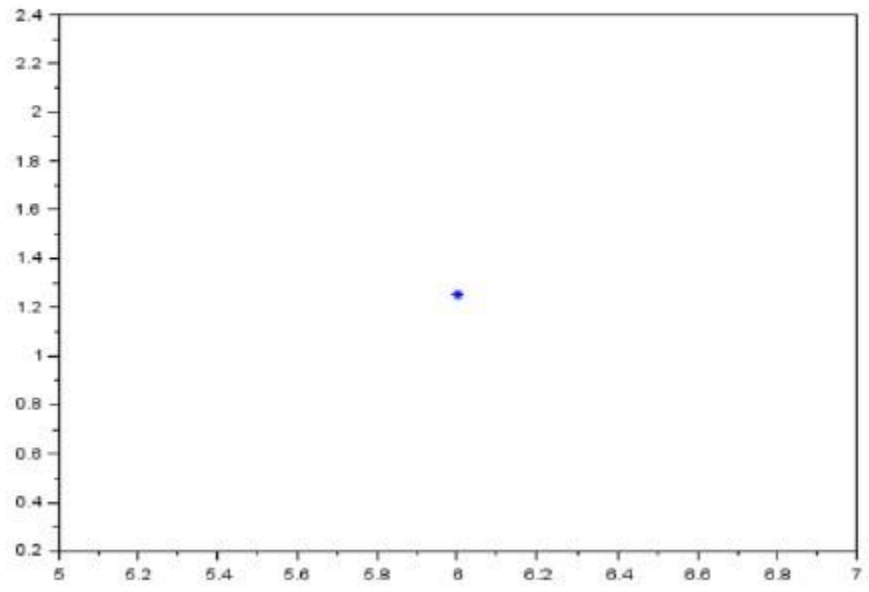
s 8.sce

```
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,"*")
```

Graphic window number 0

File Tools Edit ?

Graphic window number 0



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.

```
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)
```

Main Category : Select Main Category

Scilab code:

```
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)

ANSHUL BHARDWAJ
1/18/FET/BCS/080
```

Result:

```
 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
```

Execute Reset Report bug / Give Feedback

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 = [

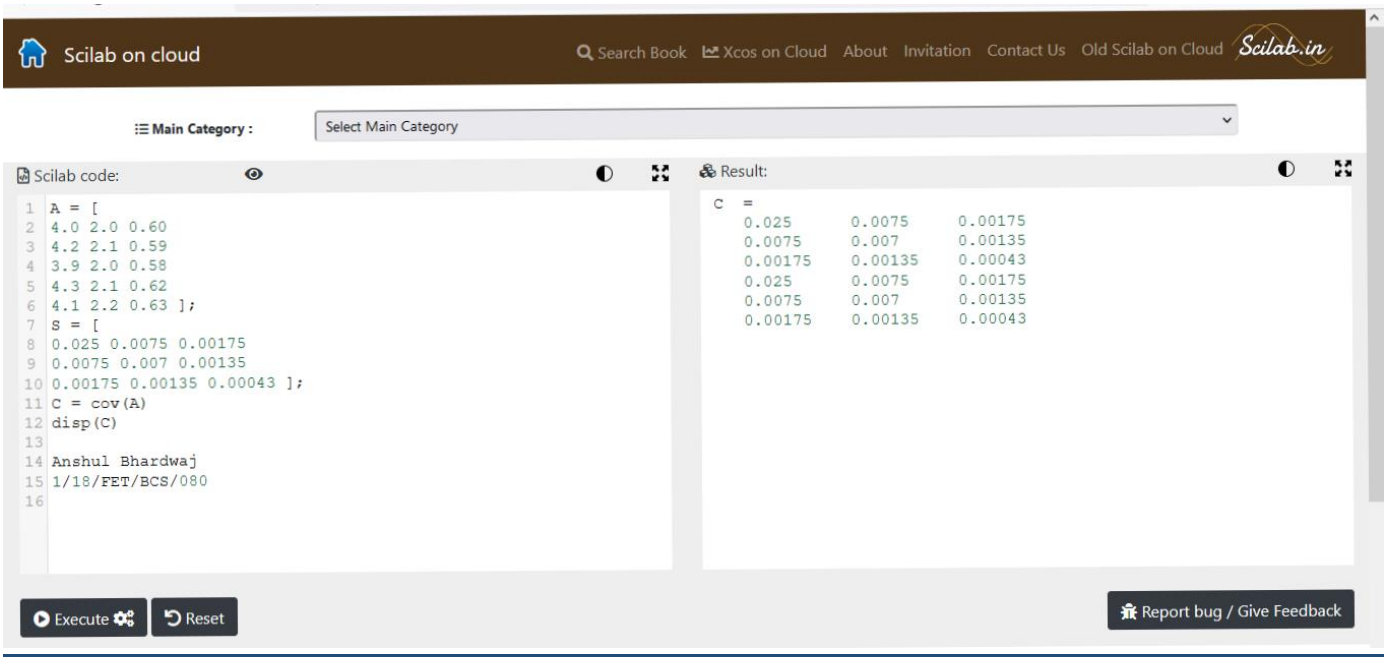
0.025 0.0075 0.00175

0.0075 0.007 0.00135

0.00175 0.00135 0.00043];

- C = cov(A)
- disp(C)

Sci lab output



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];

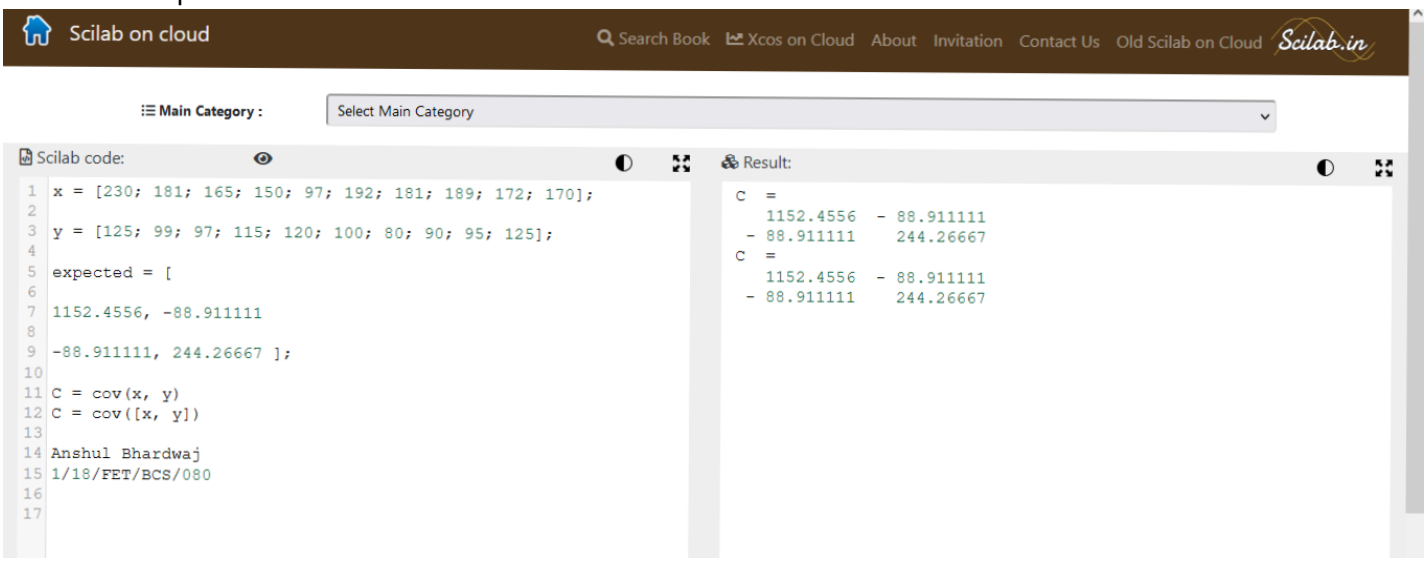
❼ expected = [

1152.4556, -88.911111

-88.911111, 244.26667];

❼ C = cov(x, y)
❼ C = cov([x, y])

Sci lab output:-



Experiment: 10(c)

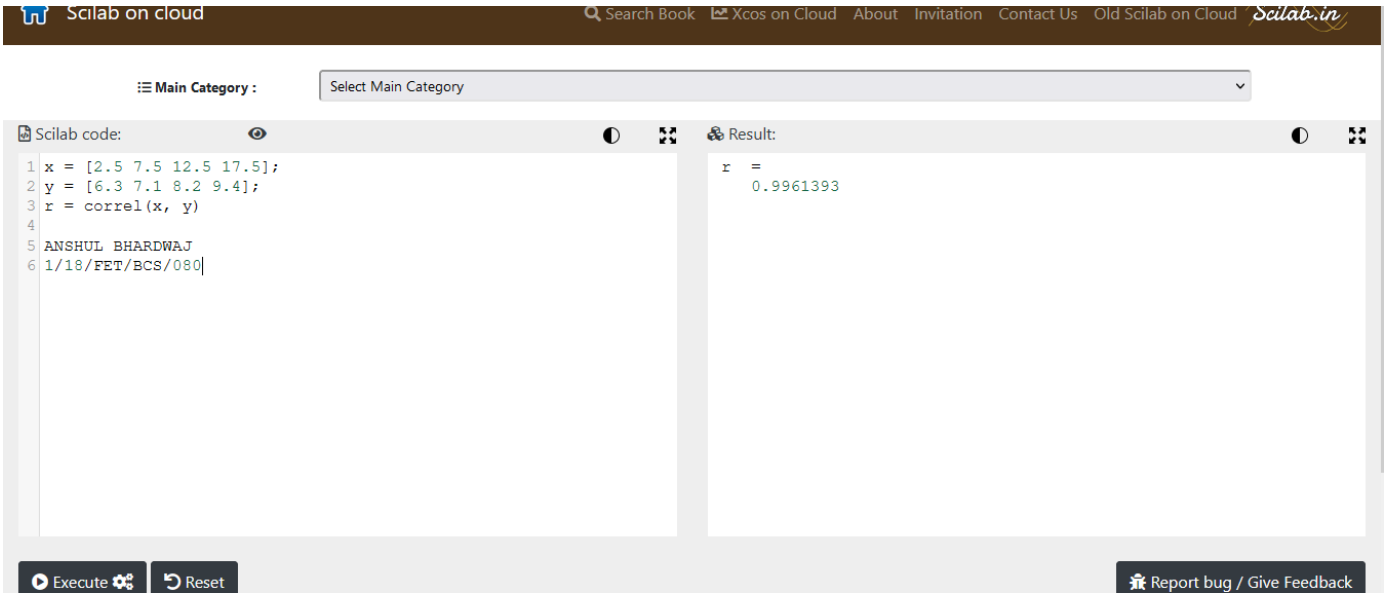
Aim: To find the Correlation

- **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.

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

Sci lab output-



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

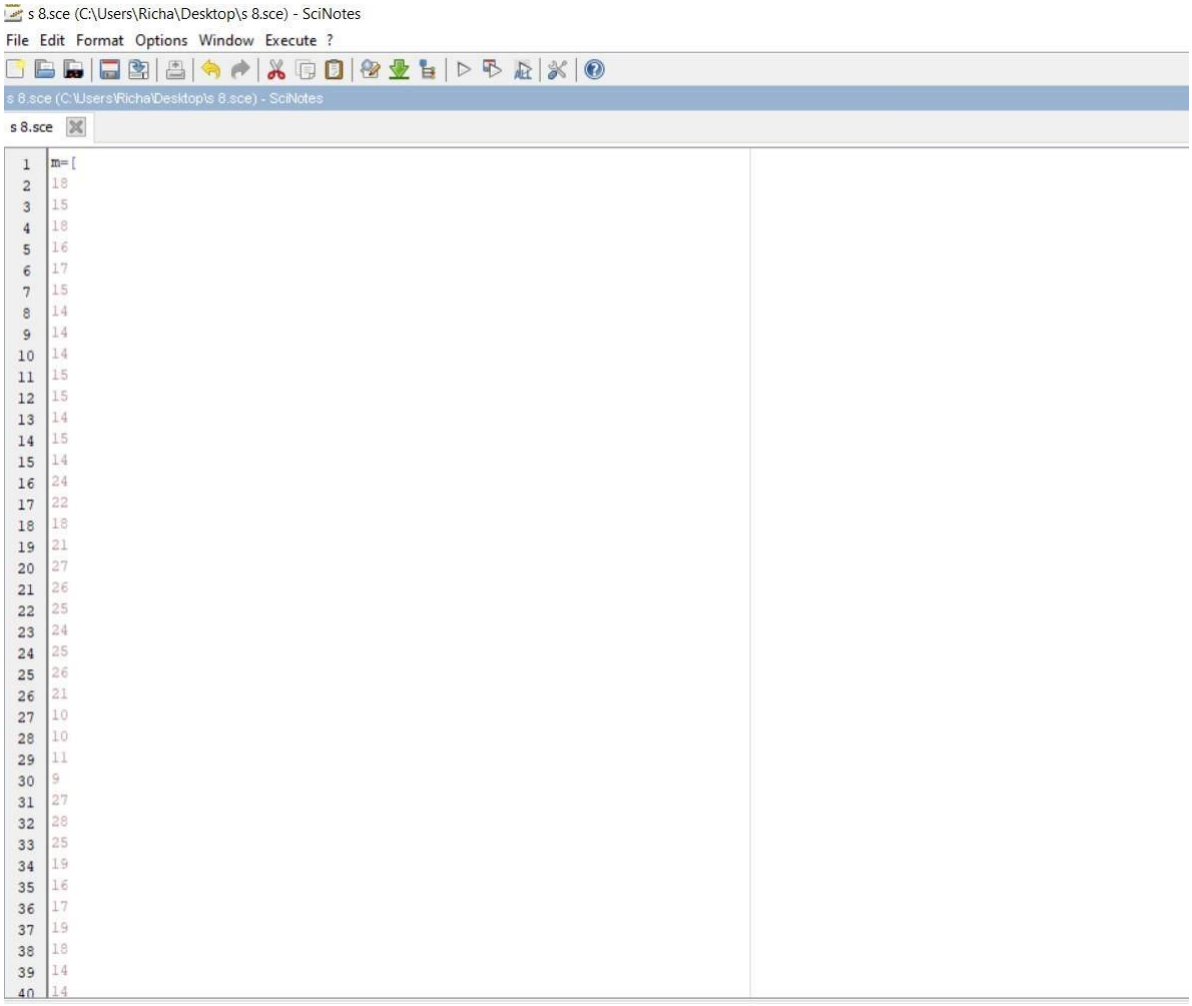
- Covariance:-

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.

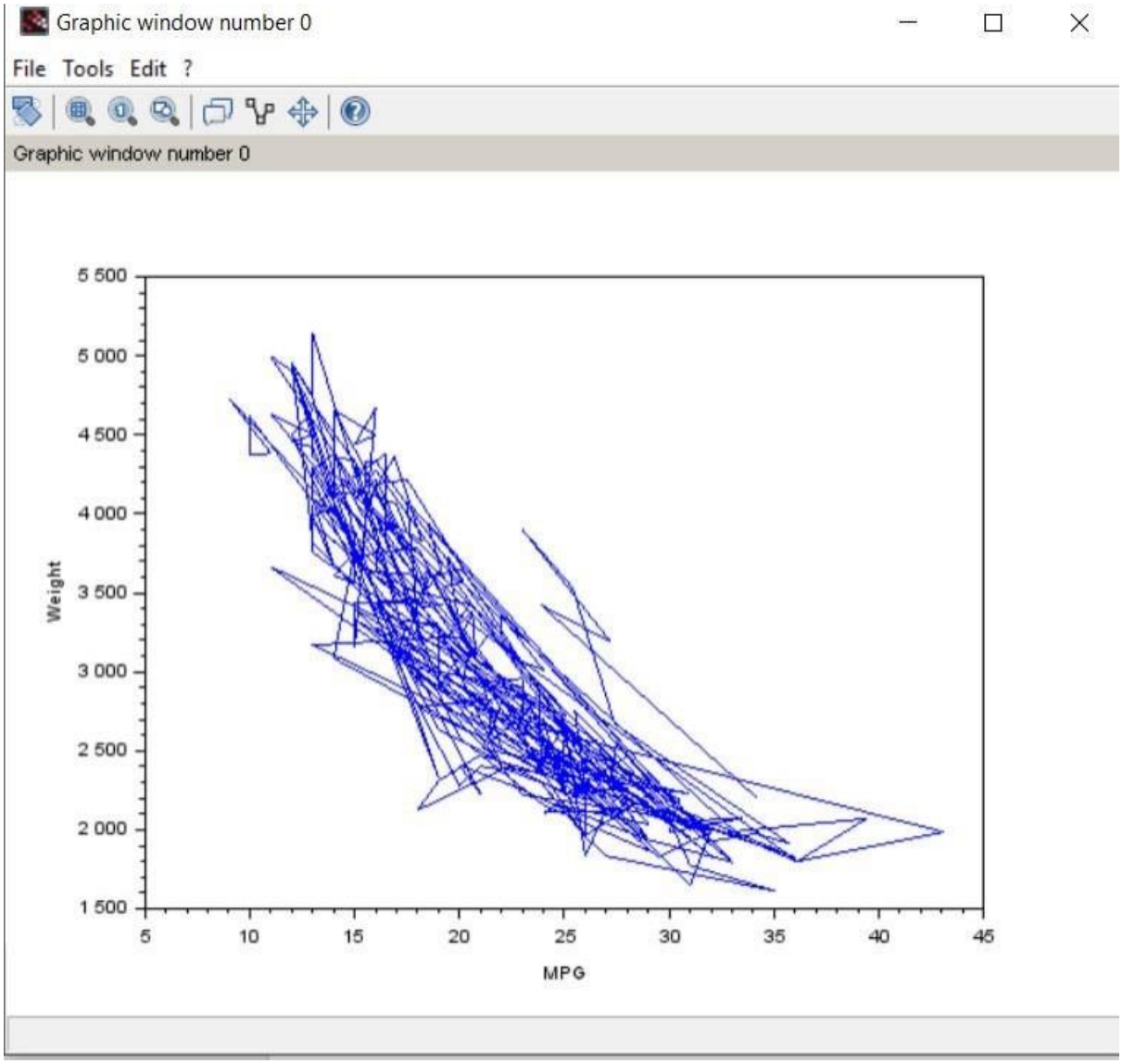
❼ m = [. . . .]

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

Sci lab output-




s 8.sce (C:\Users\Richa\Desktop\s 8.sce) - SciNotes
ile Edit Format Options Window Execute ?
s 8.sce
2330
2230
2515
2745
2555
2435
2530
3140
2795
3410
1990
2128
3245
2960
2890
3283
3380
3540
3725
3065
3030
4340
4054
3605
3940
3408
1975
1915
2670
3530
3900
3190
3420
2200
]
C=corr(m,W)
D=covrsi(m,W)
xlabel("MPG"); ylabel("Weight"); plot(m,W,'r')


Graphic window number 0
File Tools Edit ?
Graphic window number 0
5 500
5 000
4 500
4 000
3 500
3 000
2 500
2 000
1 500
Weight
5
10
15
20
25
30
35
40
45
MPG

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]

- 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

Sci lab output-

12.sce (C:\Users\Richa\Desktop\12.sce) - SciNotes

File Edit Format Options Window Execute ?

12.sce (C:\Users\Richa\Desktop\12.sce) - SciNotes

12.sce

```
m=input('enter the number of members')
j=input('enter the no 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 indeterministic')
else
    disp('the given structure is unstable')
end
```

Scilab 6.1.0 Console

```
--> editor

--> exec('C:\Users\Richa\Desktop\12.sce', -1)
enter the number of members 5

enter the no of joints 7


  "the given structure is unstable"

-->
```

Scilab 6.1.0 Console

```
--> editor

--> exec('C:\Users\Richa\Desktop\12.sce', -1)
enter the number of members 5

enter the no of joints 1


  "the given structure is indeterministic"
```

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);

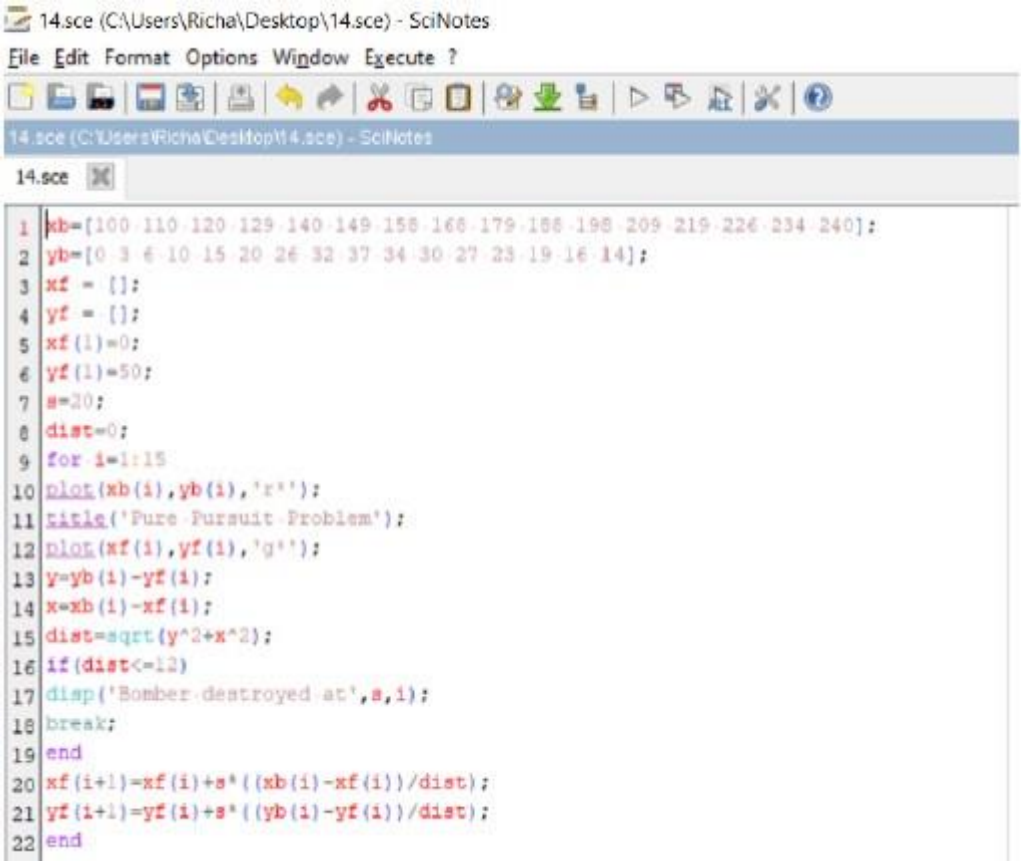
break;

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-



Scilab 6.1.0 Console

```
--> editor

--> exec('C:\Users\Richa\Desktop\14.sce', -1)

  "Bomber destroyed at"

   20.

   11.

-->
```