**LAB – 1 REPORT**

**ESE 2014 DIGITAL SIGNAL AND PROCESSING STORAGE**

**SUBMITTED BY: FRANCIS VIJITH(C0742980)**

**AKHIL GEORGE(C0748466)**

**SUBMITTED ON: 01/30/2019**

**1: CREATE A VECTOR.**

>> A = [1,2,3] //creating a row vector.

A =

1 2 3

>> B = [1;2;3] //creating a column vector.

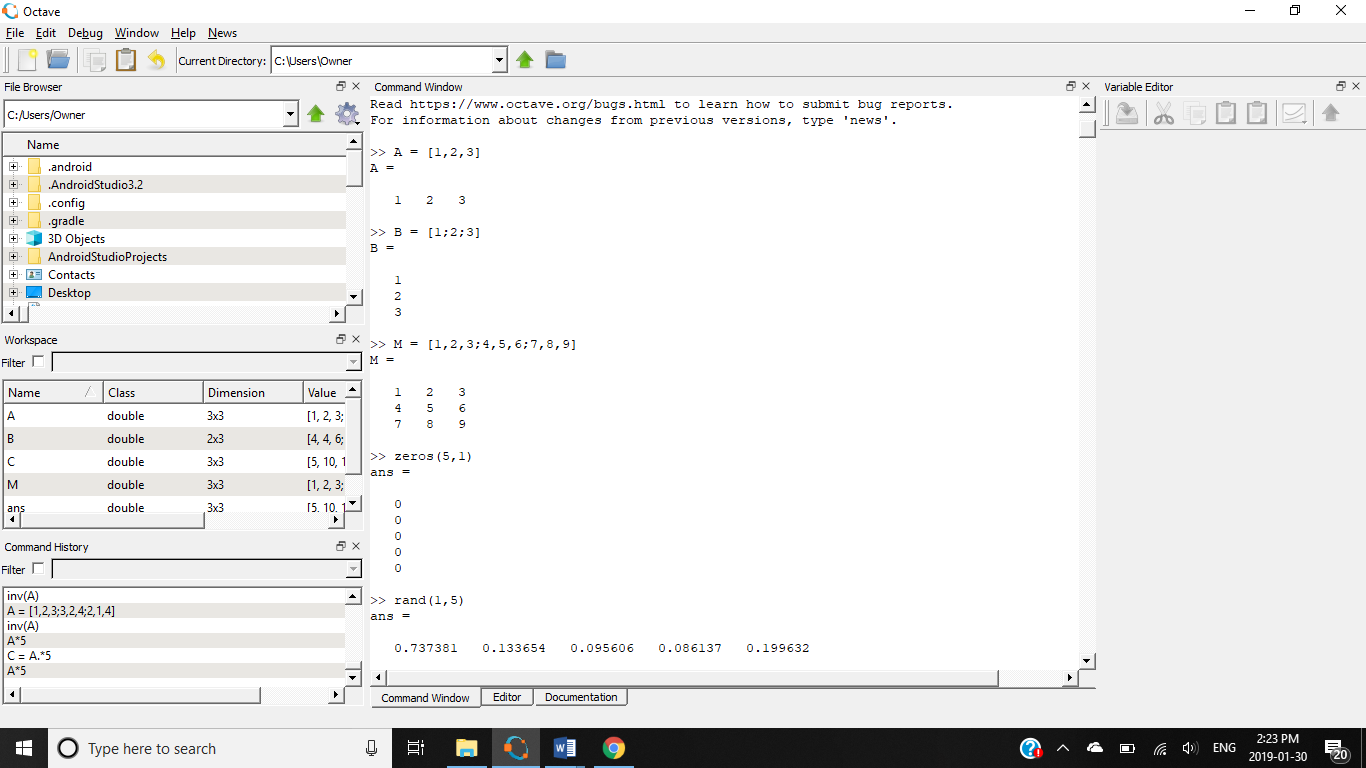
B =

1

2

3

**SCREENSHOT :**



**2: CREATE A MATRIX.**

>> M = [1,2,3;4,5,6;7,8,9] //creating a matrix.

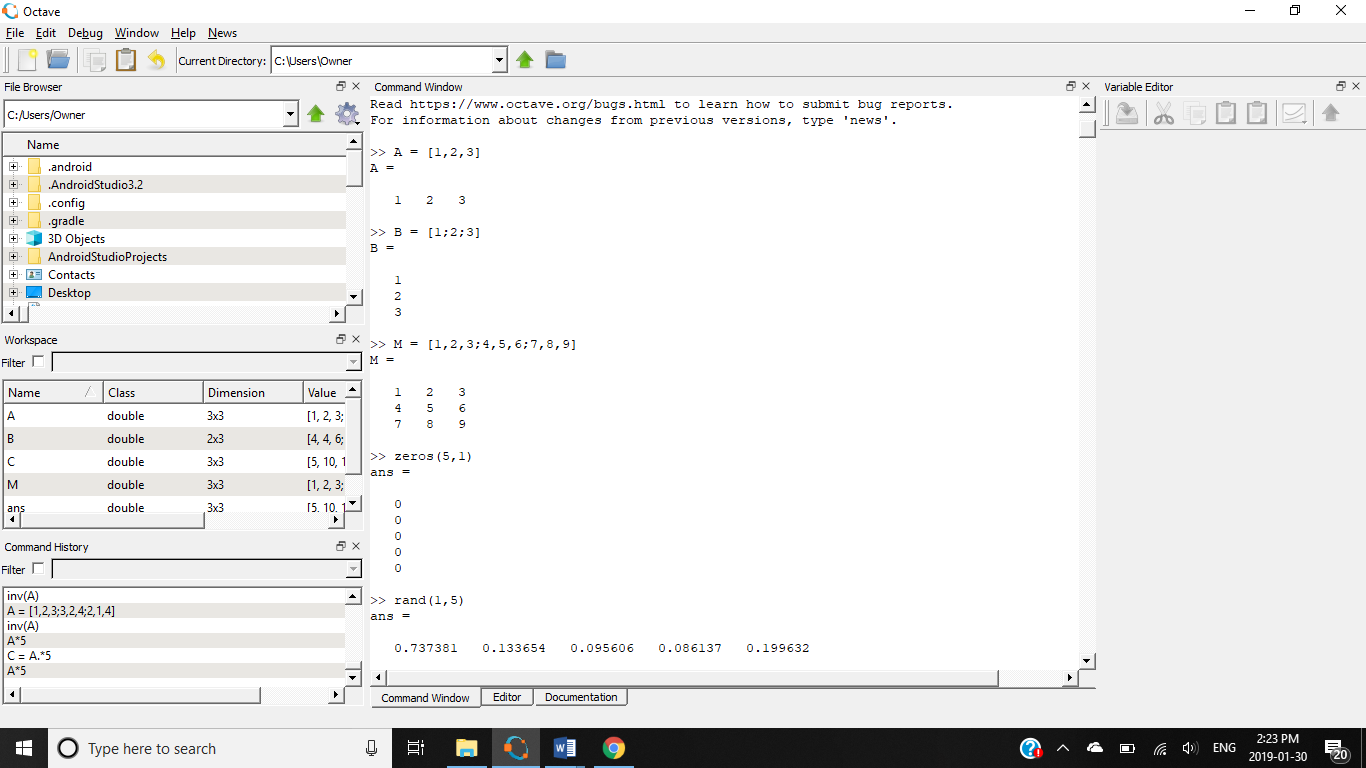
M =

1 2 3

4 5 6

7 8 9

**SCREENSHOT :**



**3. Create a 5x1 vector of zeros. Create a 1x5 vector of random numbers.**

>> zeros(5,1) //creating a vector of zeroes

ans =

0

0

0

0

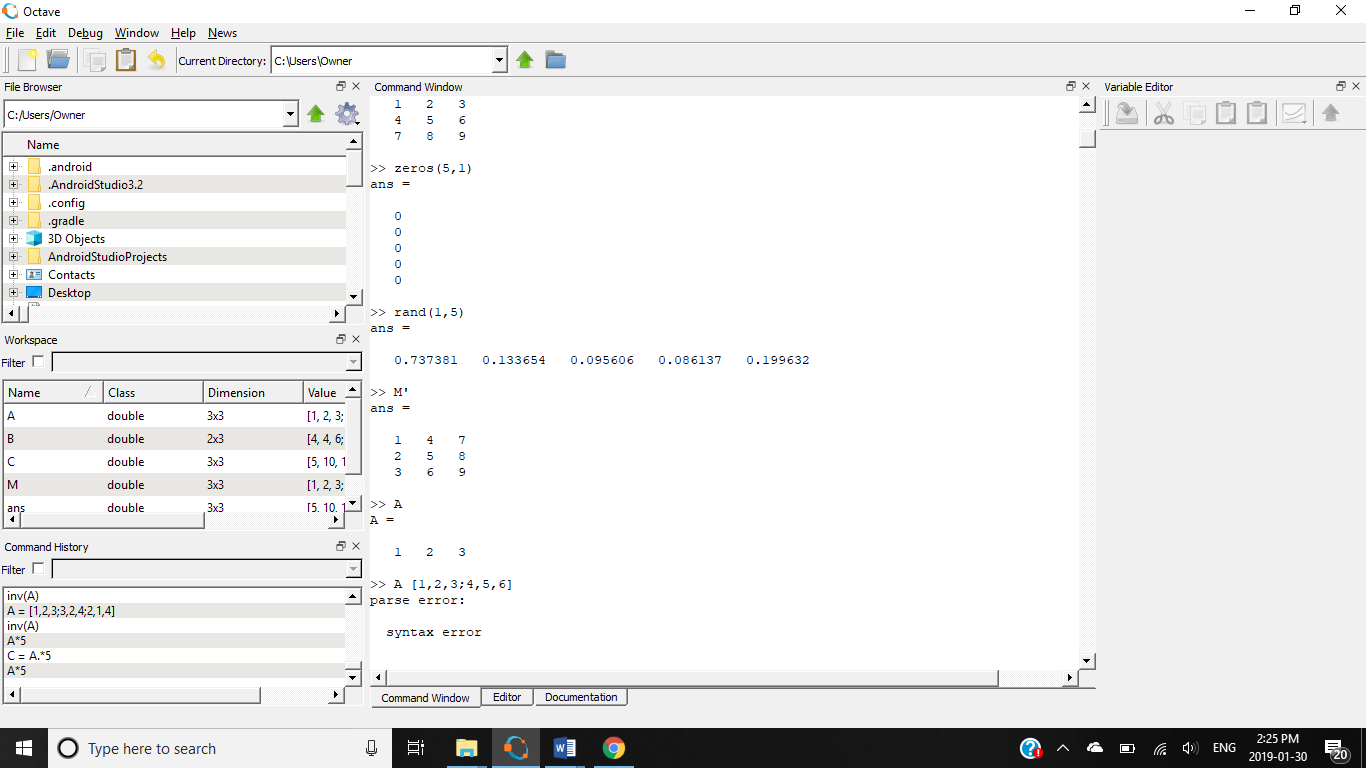
0

>> rand(1,5) //creating a vector of random numbers

ans =

0.737381 0.133654 0.095606 0.086137 0.199632

**SCREENSHOT :**



**4. Transpose a matrix.**

>> M' //transposing a matrix

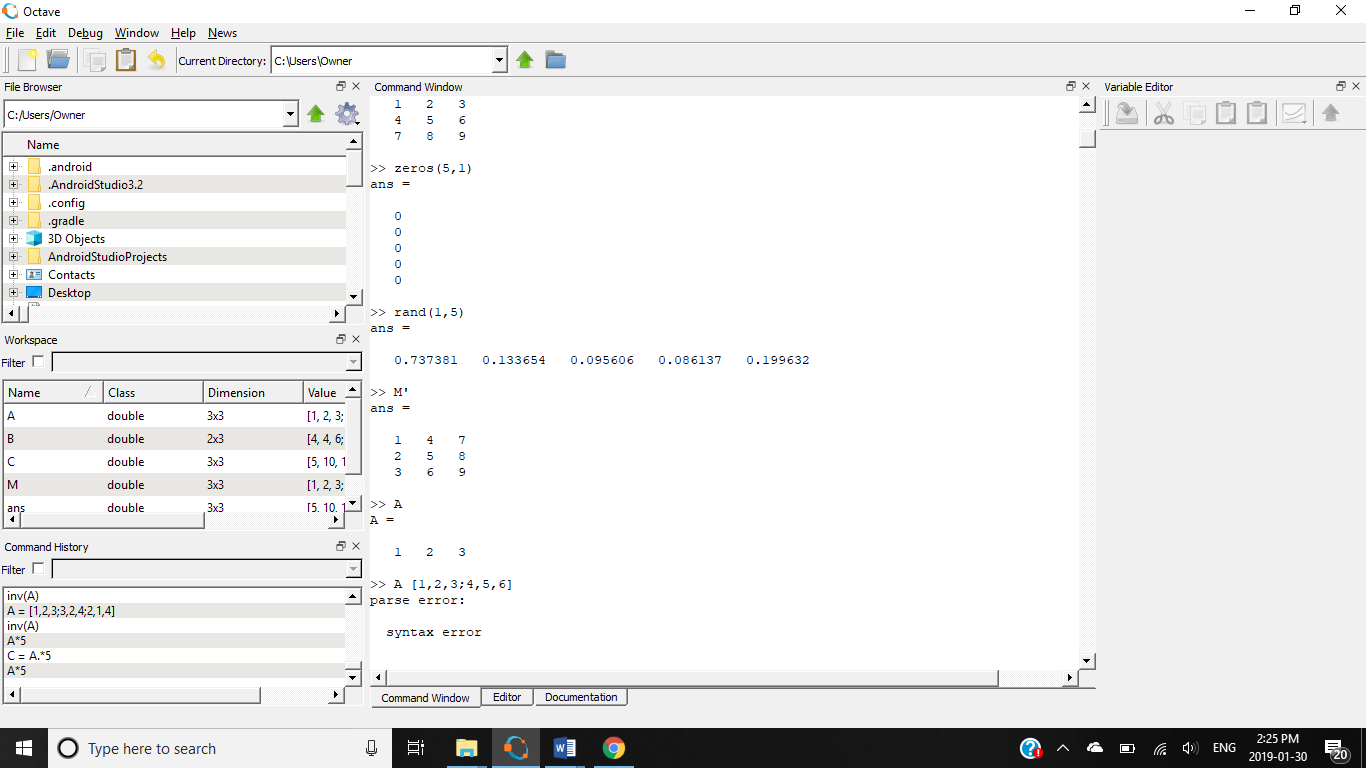
ans =

1 4 7

2 5 8

3 6 9

**SCREENSHOT :**



**5. Compute the inner product of two matrices. Compute the cross product of two matrices. Compute the inverse of a matrix.**

>>A = //variable A

1 2 3

4 5 6

>> B = //variable B

4 4 6

3 4 5

>> dot(A,B) //dot product of A and B

ans =

16 28 48

>> cross(A,B) //cross product of A and B

ans =

0 6 -4

1 -2 1

A = //square matrix A

1 2 3

3 2 4

2 1 4

>> inv(A) //inverse of A

ans =

-0.57143 0.71429 -0.28571

0.57143 0.28571 -0.71429

0.14286 -0.42857 0.57143

**6. Compute the element wise multiplication of a matrix and a scalar.**

>> A\*5 //element wise scalar multiplication

ans =

5 10 15

15 10 20

10 5 20

>> A.\*C //element wise matrix multiplication

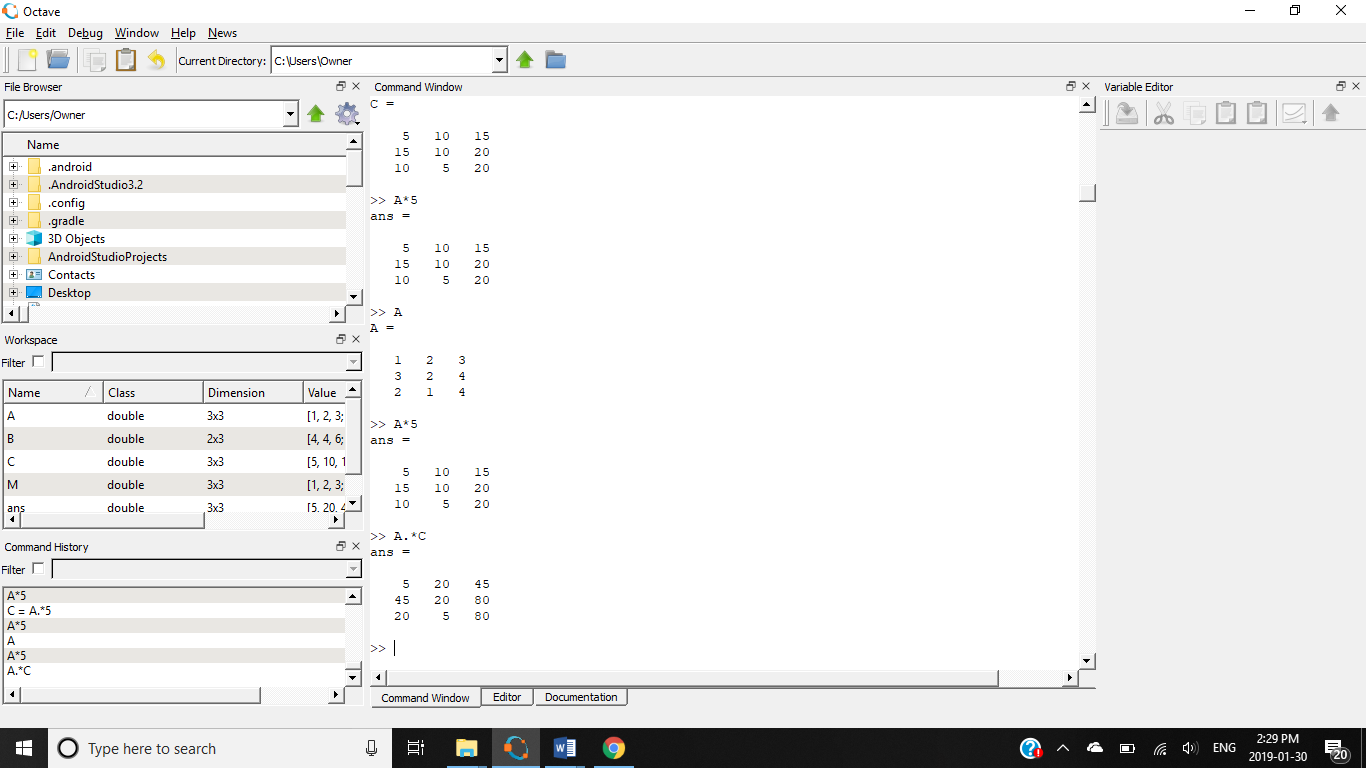
ans =

5 20 45

45 20 80

20 5 80

**SCREENSHOT :**



**7. Concatenate two matrices.**

>> [A C] //concatenating matrices A and C

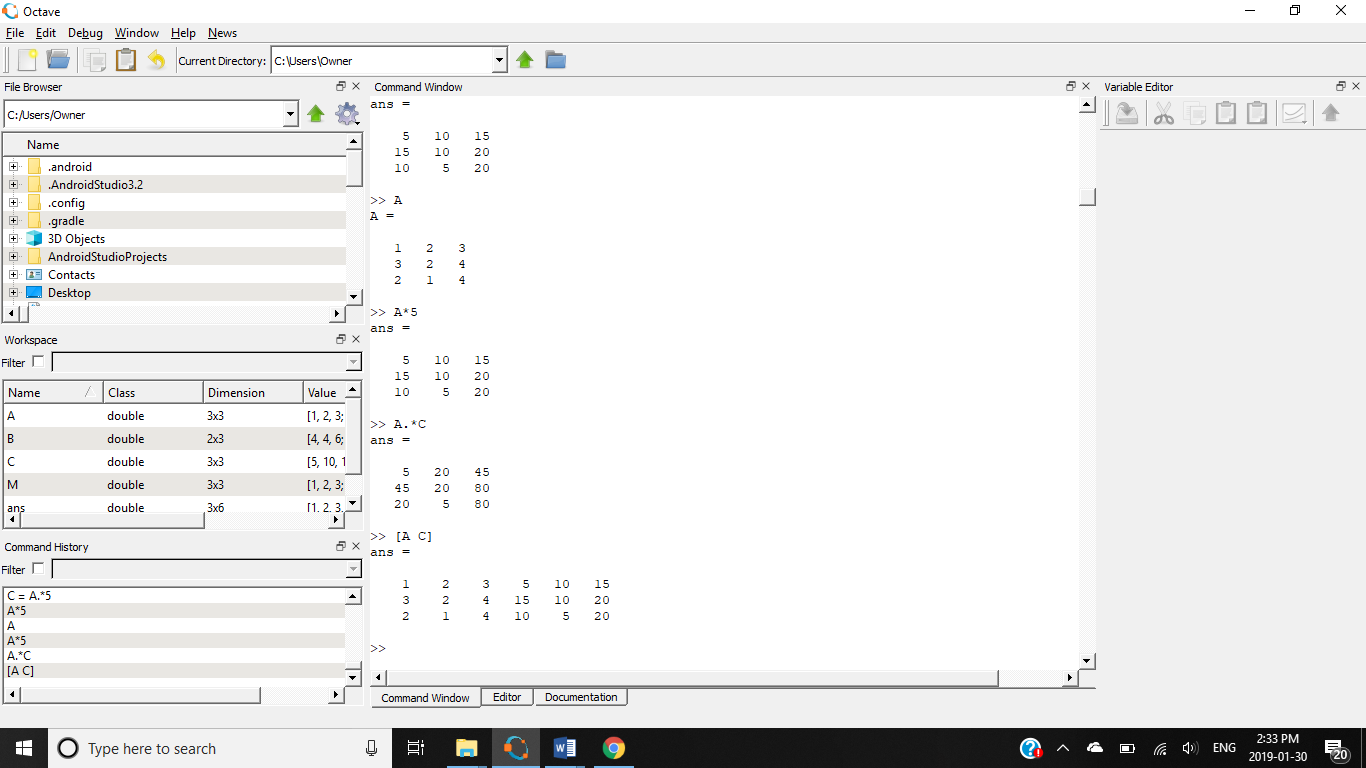
ans =

1 2 3 5 10 15

3 2 4 15 10 20

2 1 4 10 5 20

**SCREENSHOT:**



**8. Create a vector of complex numbers.**

>> A = [1,2,3] //creating vector A

A =

1 2 3

>> B = [4,5,6 ]//creating vector B

B =

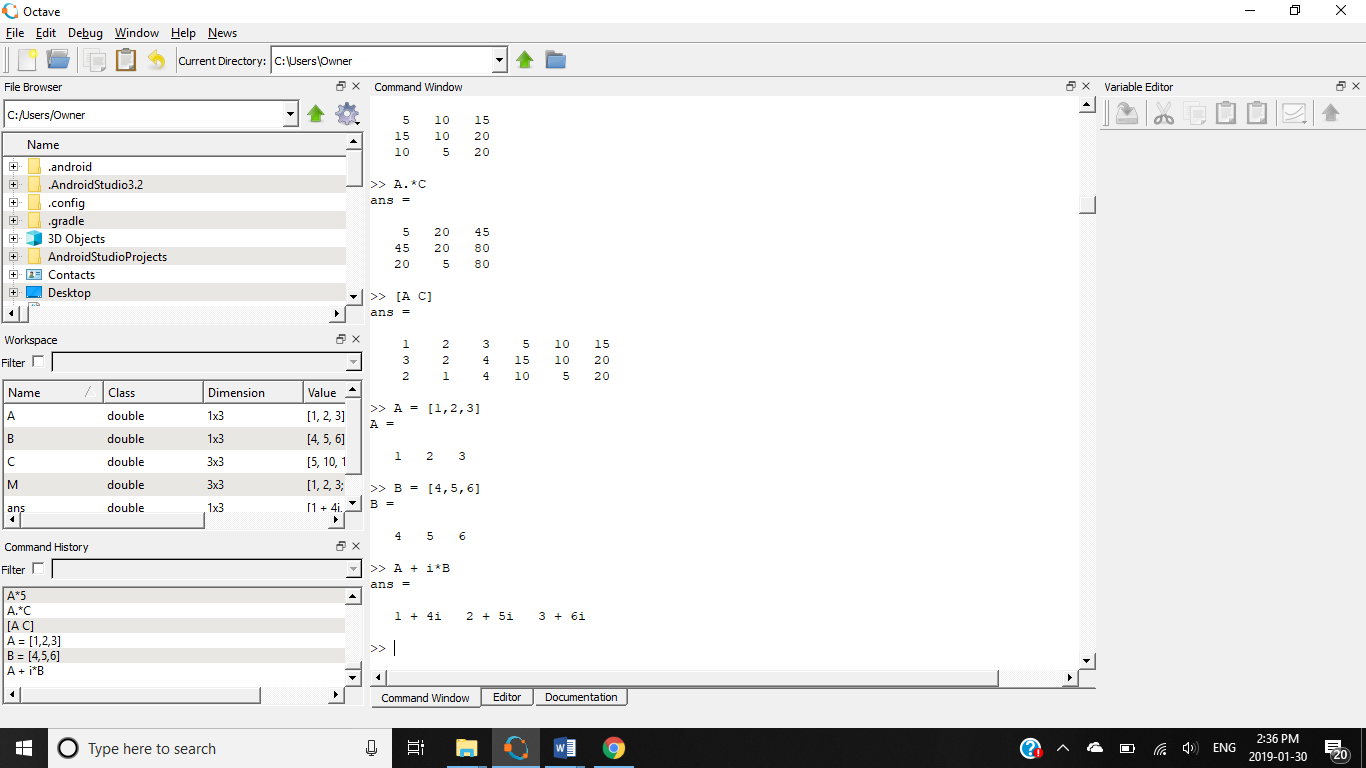
4 5 6

>> A + i\*B //forming a vector of complex numbers

ans =

1 + 4i 2 + 5i 3 + 6i

**SCREENSHOT:**



**9. Multiply a row of a matrix with an element of that same matrix.**

>> A //creating a matrix A

A =

1 2 3

4 5 6

7 8 9

>> A(1,:)= A(1,:)\*A(3,2) //multiplying the first row with A(3,2)=8

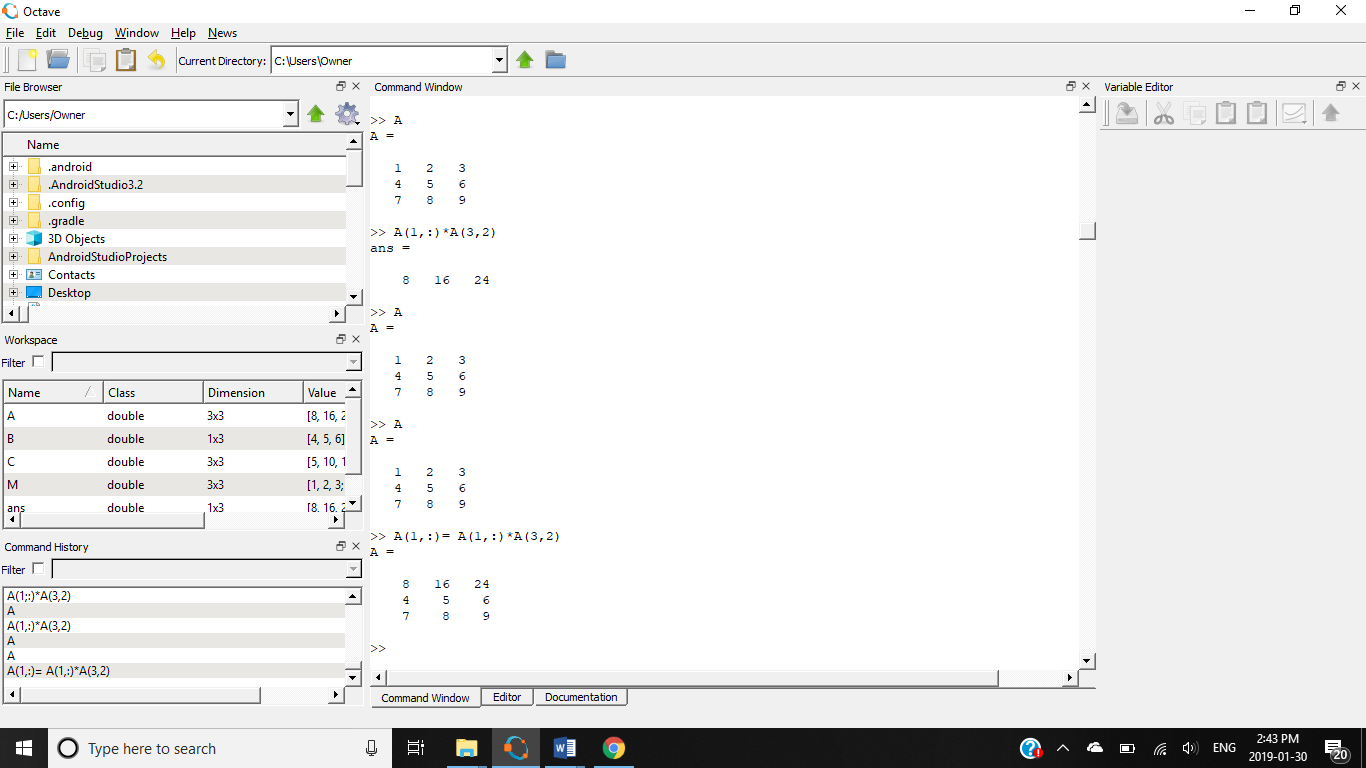
A =

8 16 24

4 5 6

7 8 9

**SCREENSHOT:**



**10.Generate a vector of values ranging from 0 to 500 with 100 elements.**

>> B = linspace(0,500,100)

B =

Columns 1 through 8:

0.00000 5.05051 10.10101 15.15152 20.20202 25.25253 30.30303 35.35354

Columns 9 through 16:

40.40404 45.45455 50.50505 55.55556 60.60606 65.65657 70.70707 75.75758

Columns 17 through 24:

80.80808 85.85859 90.90909 95.95960 101.01010 106.06061 111.11111 116.16162

Columns 25 through 32:

121.21212 126.26263 131.31313 136.36364 141.41414 146.46465 151.51515 156.56566

Columns 33 through 40:

161.61616 166.66667 171.71717 176.76768 181.81818 186.86869 191.91919 196.96970

Columns 41 through 48:

202.02020 207.07071 212.12121 217.17172 222.22222 227.27273 232.32323 237.37374

Columns 49 through 56:

242.42424 247.47475 252.52525 257.57576 262.62626 267.67677 272.72727 277.77778

Columns 57 through 64:

282.82828 287.87879 292.92929 297.97980 303.03030 308.08081 313.13131 318.18182

Columns 65 through 72:

323.23232 328.28283 333.33333 338.38384 343.43434 348.48485 353.53535 358.58586

Columns 73 through 80:

363.63636 368.68687 373.73737 378.78788 383.83838 388.88889 393.93939 398.98990

Columns 81 through 88:

404.04040 409.09091 414.14141 419.19192 424.24242 429.29293 434.34343 439.39394

Columns 89 through 96:

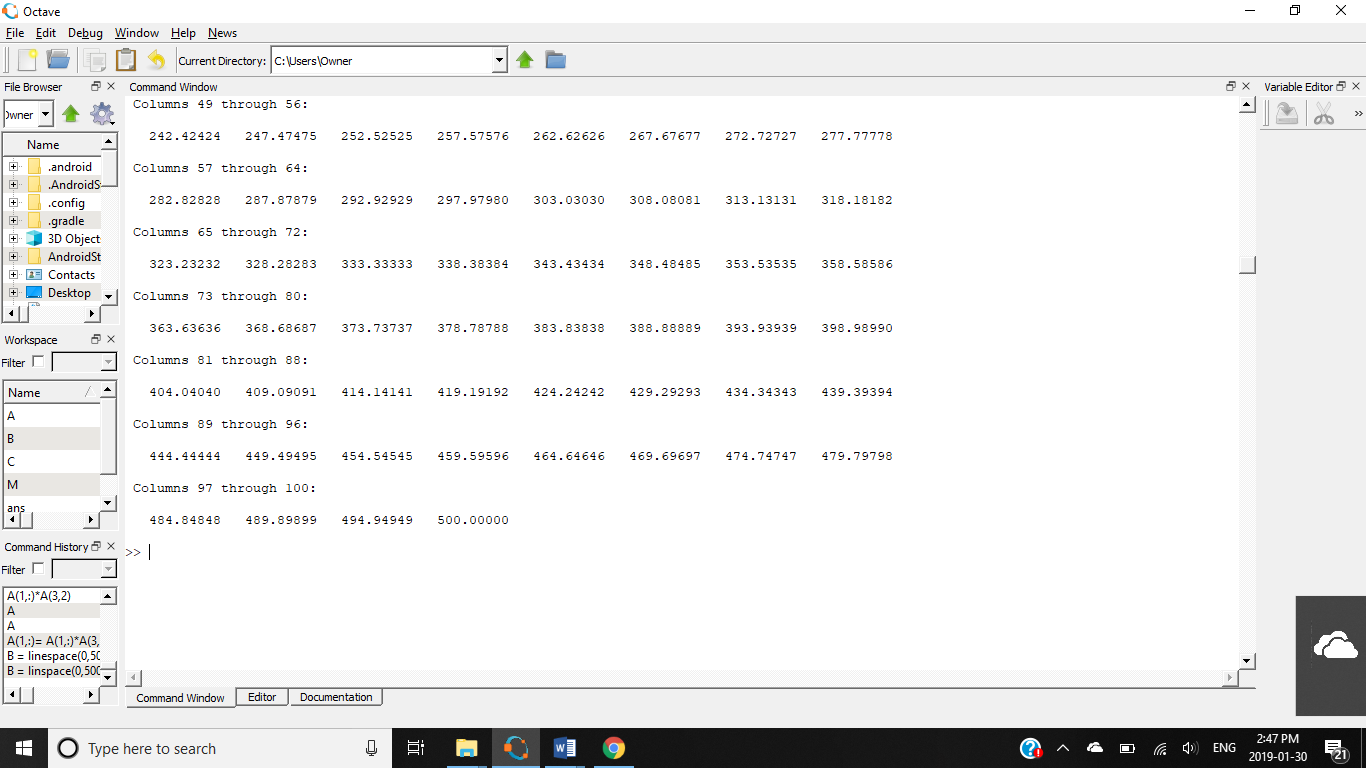
444.44444 449.49495 454.54545 459.59596 464.64646 469.69697 474.74747 479.79798

Columns 97 through 100:

484.84848 489.89899 494.94949 500.00000

**SCREENSHOT:**

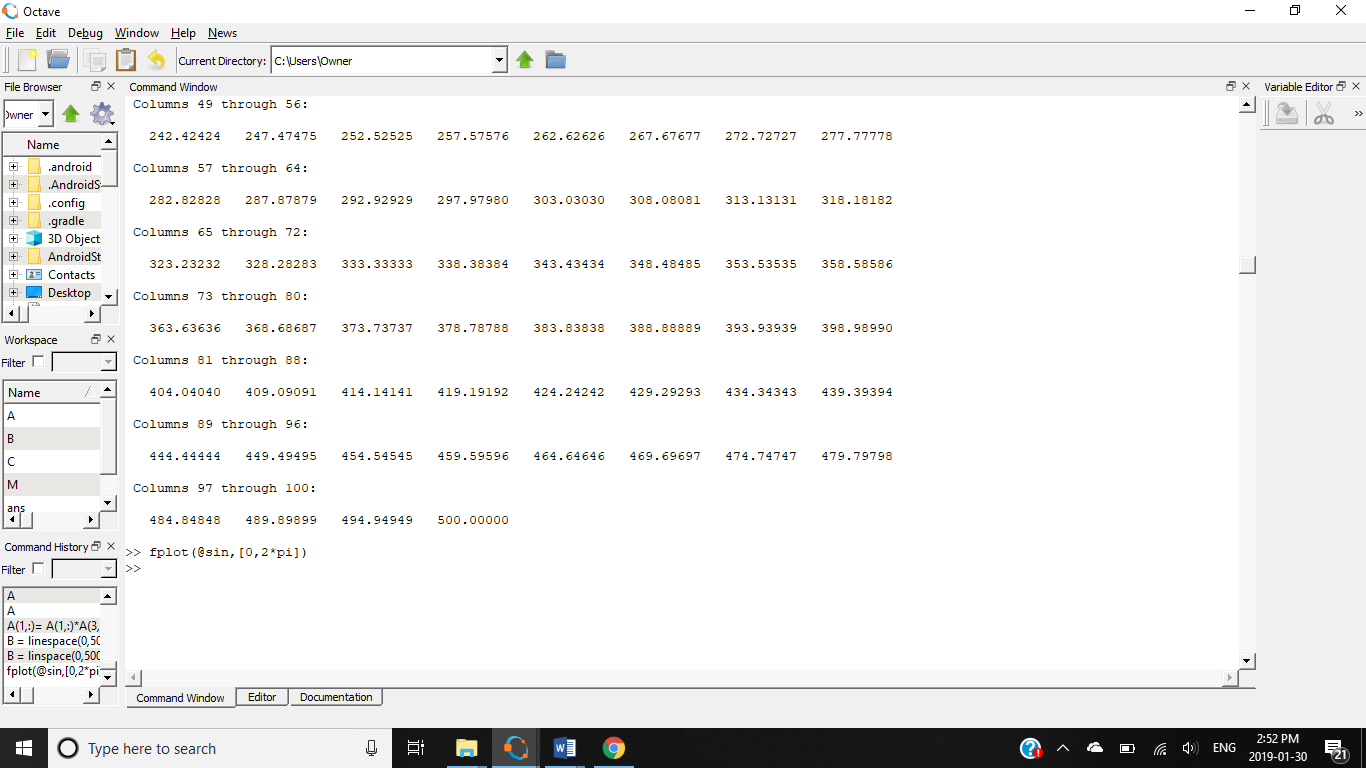


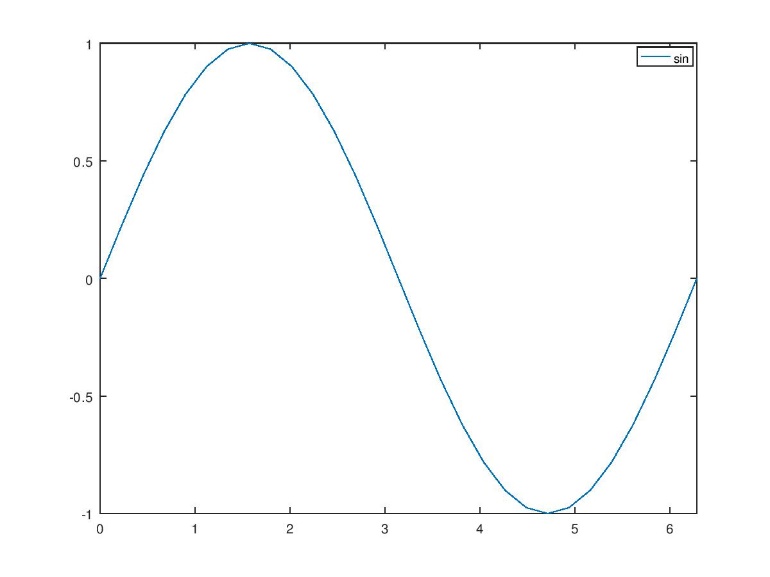


**11.Create a 2D plot of the sine function between 0 and 2π.**

fplot(@sin,[0,2\*pi])

**SCREENSHOT:**





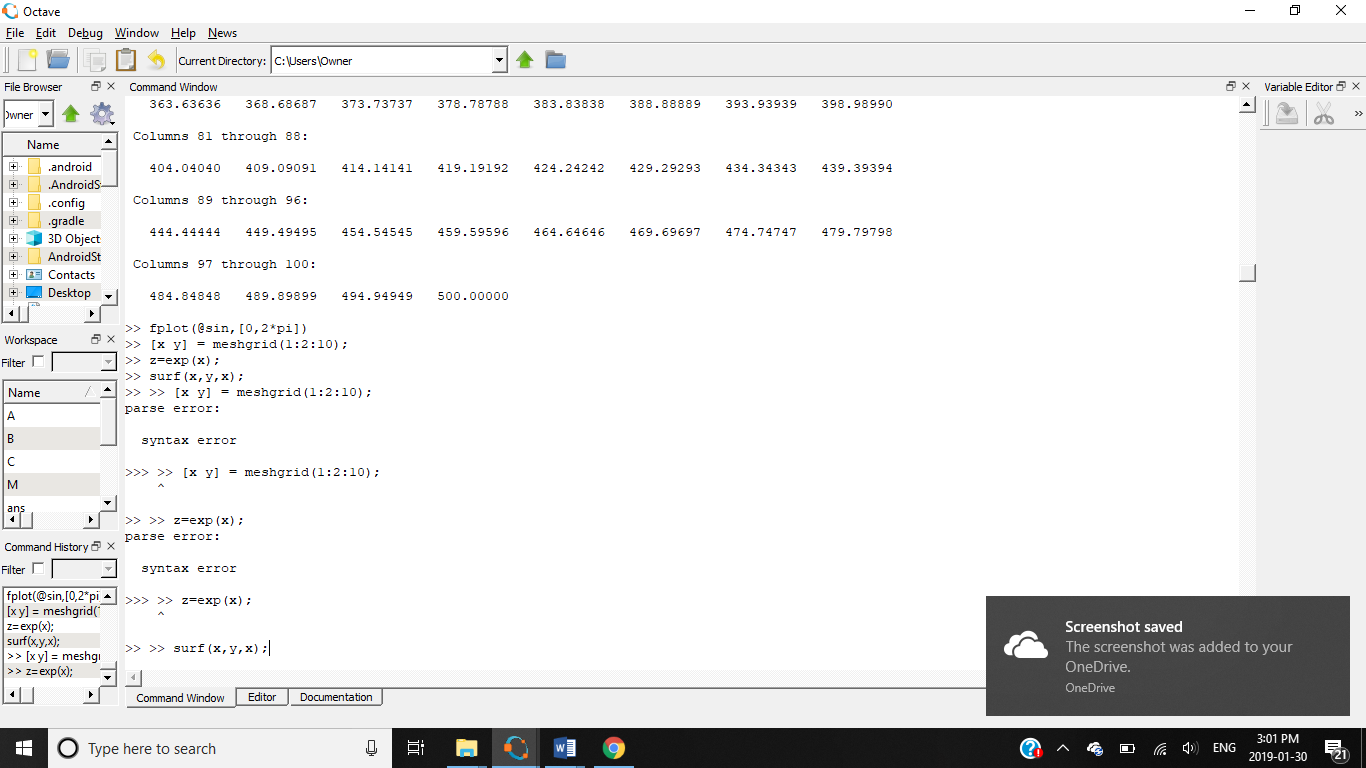
**12.Create a 3D plot of a surface by creating a grid along the X and Y axes and plotting the Z-coordinate according to the exponential function.**

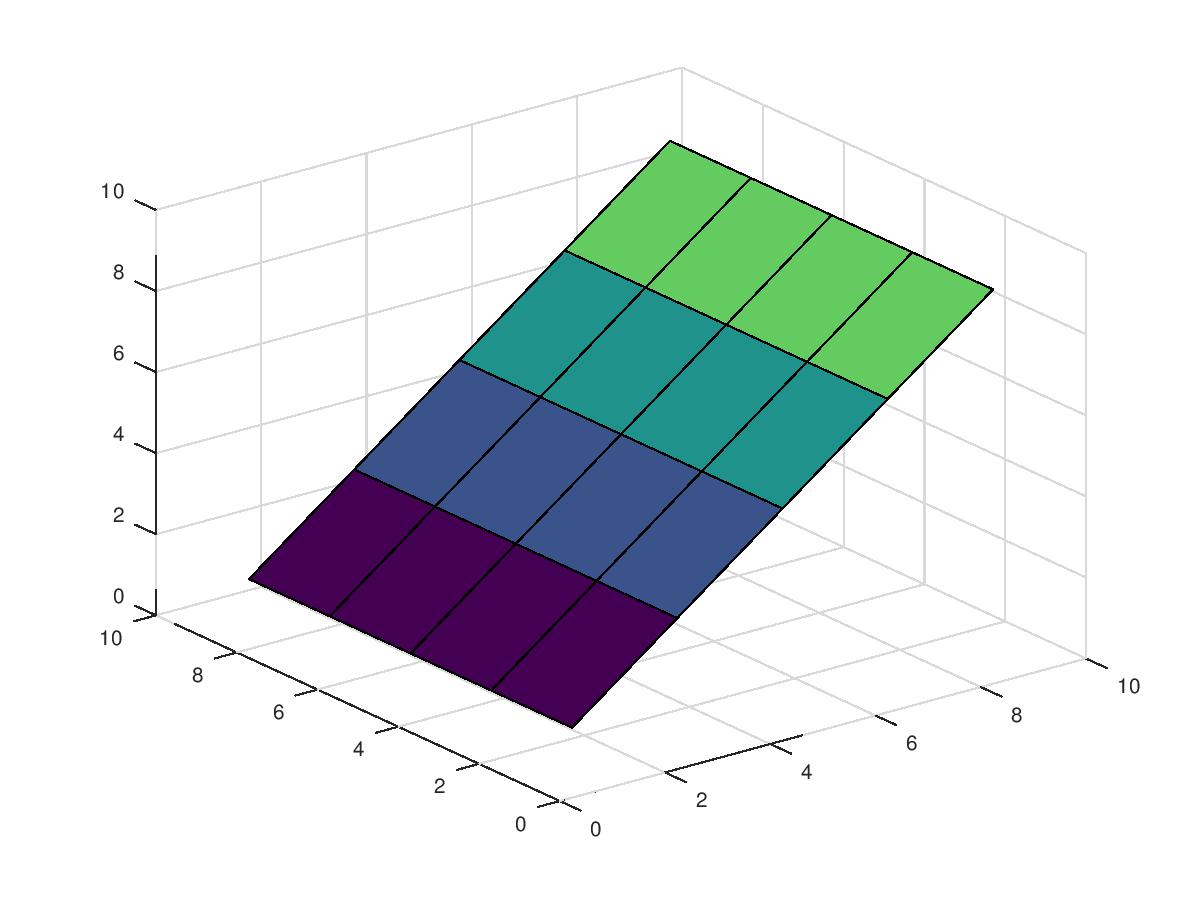
>> [x y] = meshgrid(1:2:10); //plotting the x-y values

>> z=exp(x); //giving exponential of x to z

>> surf(x,y,x); //plotting a 3d graph

**SCREENSHOT:**





**13.Write a script to plot a vector of random data. Draw a horizontal line at the mean. Save the script and run it from the command line.**

x=1:10; //script and giving values for x axis

y=x+randn(1,10); //giving values for y axis

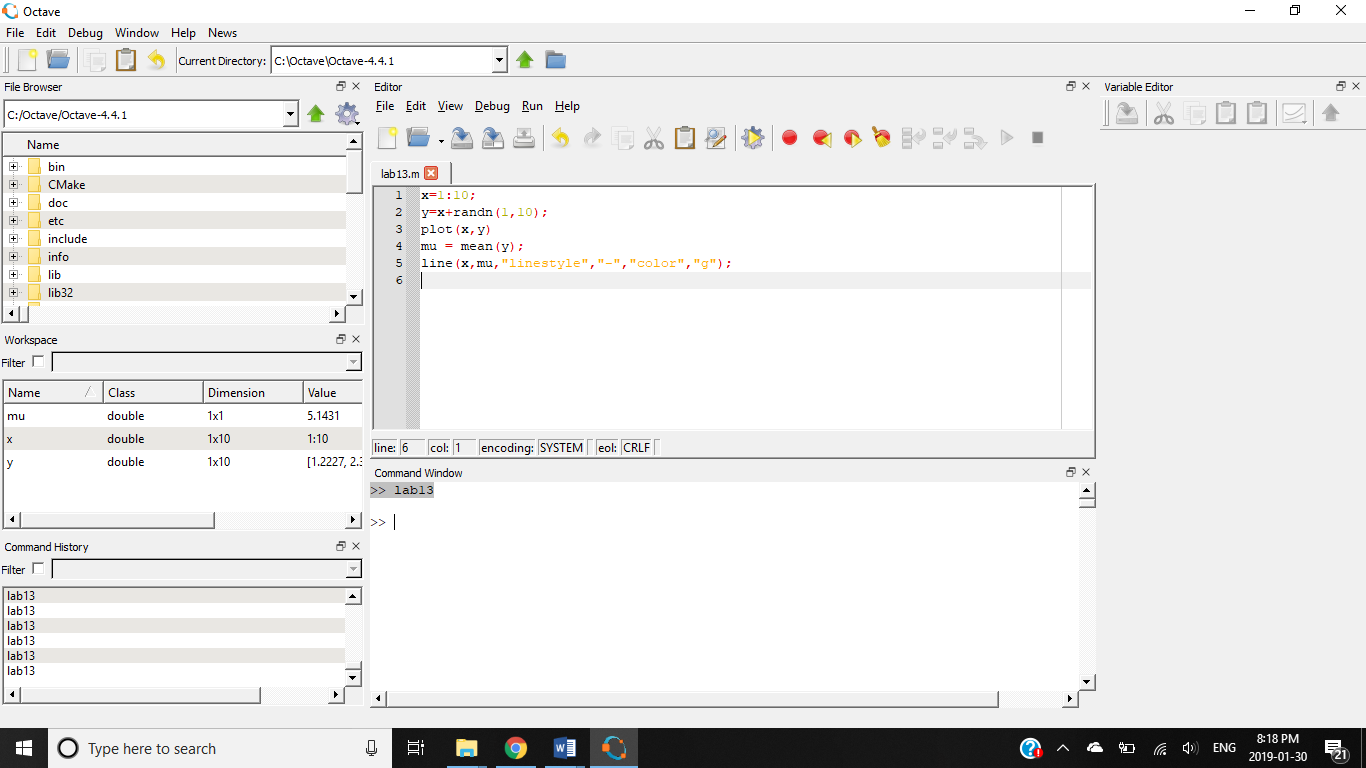
plot(x,y) //plotting x and y axis

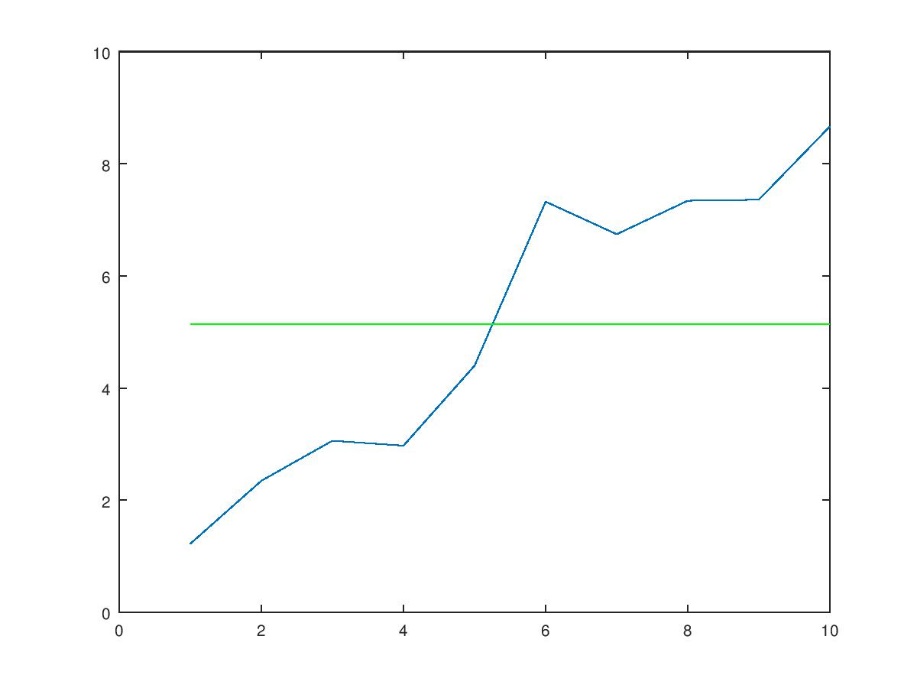
mu = mean(y); //finding mean value for y values

line(x,mu,"linestyle","-","color","g"); //editing the graph properties

>> lab13 //command window

**SCREENSHOT:**





**14.Write another script that calculates the mean of five samples of data from a vector of random data. Calculate the overall mean. Use a for loop to perform the calculations. For each iteration of the loop print out the intermediate results. Use an if..else control block to display the results depending on whether the mean of the samples is less than, greater than or equal to the overall mean.**

**SCRIPT:**

A = 5;

B = 35;

for i=1:A

Sampleno=[int2str(i)]

sample=rand(1,5)

disp(C)

endfor

C=mean(sample)

if C<0.49

disp('mean is less')

elseif C>0.51

disp('mean is greater')

else

disp('mean is unexpected')

end

**COMMAND WINDOW:**

>> lab13

Sampleno = 1

sample =

0.61741 0.61568 0.68803 0.43065 0.74605

0.43734

Sampleno = 2

sample =

0.15938 0.87189 0.14976 0.63004 0.20204

0.43734

Sampleno = 3

sample =

0.107845 0.449667 0.505295 0.052070 0.173763

0.43734

Sampleno = 4

sample =

0.88697 0.19188 0.13675 0.28365 0.21716

0.43734

Sampleno = 5

sample =

0.85796 0.64379 0.54713 0.92689 0.47972

0.43734

C = 0.69110

mean is greater

>>

**15.Create a function that calculates the sum of an arbitrary number of sinusoidal terms. Call this function from the MATLAB command line or in a MATLAB script (.m).**

function [c]=arbit\_sin()

a=linspace(0,2\*pi,750);

b=sin(a);

c=sum(b);

endfunction

