++++++++++++++++++ Lab 005 ++++++++++++++++++

++++++++++++++++++ 2154638 ++++++++++++++++++

# Mathematical Skills for Data Scientists
# Lab Exercises 5

CSCM70

## Lab 5 Solutions

Lab Exercise Sheet 5 Start Assignment

**Due** Monday by 23:59 **Points** 4 **Submitting** a file upload **File types** m, pdf, txt, and xls **Available** 7 Nov at 11:00 - 15 Nov at 2:00

You can find the second exercise sheet below. You have a full week to complete it.

Lab_ExerciseSheet5.pdf

Please upload the file as

- matlab m file (not live script mlx)
- text
- pdf
- xls

If you are doing multiple submissions, please include all the files in each submission. We will only mark the last submission (and the files attached to it)

◂ Previous Next ▸

**Question 1: ->**

**Exercise 1** (Excel). Do linear regression on the following data:

| x | y |
|---|---|
| 1.1 | 2.7 |
| 1.2 | 2.8 |
| 2.4 | 5.2 |
| 3.1 | 6.8 |

What values for slope and intercept do you get? Try out forcing the intercept to be zero. Does the slope change? If so, why? 1 mark

## Answer 1:

lab5_answer1.xlsx

Excel Sheet above.

**Description :**
In the below charts we've plotted the data given in question. The data is plotted for linear equation which can be represented as $y = mx + c$, with $m$ as the slope if the line and $c$ as the intercept of the line(the value of y where x = 0)

**Chart 1** → Shows the best fitting line

- B8 is calculated as =INDEX(LINEST(C2:C5,B2:B5),1)
- C8 is calculated as =INDEX(LINEST(C2:C5,B2:B5),2)
- Dashed line represents extrapolation to the x and y axis

**Chart 2** → shows the line where we've set the intercept as Zero

- B22 is calculated as =INDEX(LINEST(C2:C5,B2:B5,0),1)
- C22 is calculated as =INDEX(LINEST(C2:C5,B2:B5,0),2)
- Dashed line represents extrapolation to the x and y axis where it can be seen that y intercept is zero.

# OUTPUT EXCEL:

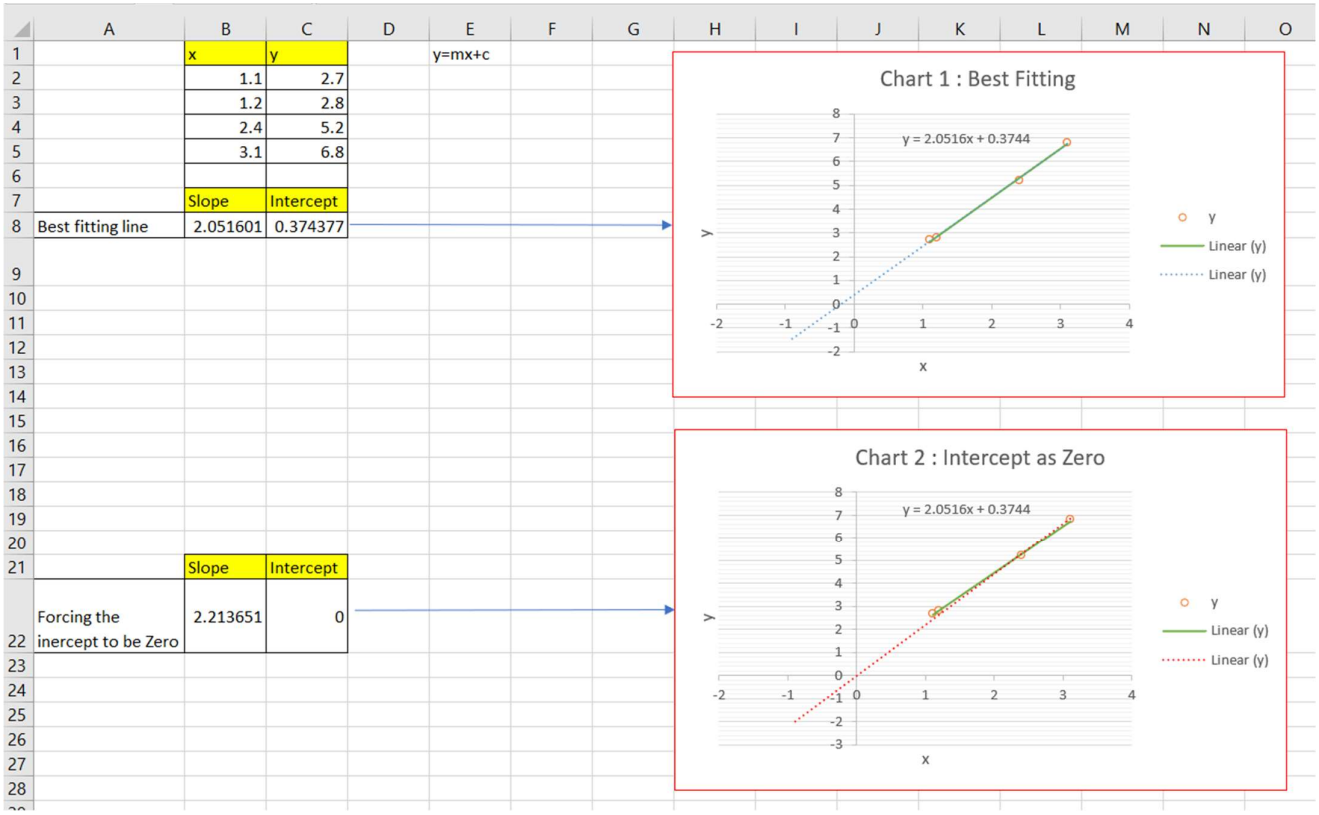



Image : Excel screenshot for Chart 1 and Chart 2

Yes, the slope changes as we tried to change the slope of the line by changing the intercept to be zero. It is also called regression through origin; we need this to interpret the real-world situation where most of the things starts from origin [3]. In other words, we are creating the variance through cluster sampling.

**Question 2: ->**

**Exercise 2** (Matlab). Write a Matlab function that take a data matrix to find the covariance matrix (Input: data matrix and output: covariance of data matrix). You need to use the formula (in matrix form) discussed in the lectures (and not inbuilt command, cov). Use this to find the covariance matrix of the data matrix.

$$\begin{pmatrix} 1.1 & 2.7 & 3.2 & 2.1 \\ 1.2 & 2.8 & 3.4 & 1.8 \\ 2.4 & 5.2 & 1.9 & 2.5 \\ 3.1 & 6.8 & 3.5 & 2 \end{pmatrix}$$

## Answer 2:

F = [1.1 2.7 3.2 2.1; 1.2 2.8 3.4 1.8; 2.4 5.2 1.9 2.5; 3.1 6.8 3.5 2]

cov_manual(F)

cov_manual.m

Without using the built-in function, we are finding the covariance of the data matrix F , using the formula discussed in class.

$F$ be the data matrix. Then $F - 1_N \cdot \bar{X}$ is the normalized data matrix. Now

$$Q = \frac{1}{N-1}(F - 1_N \cdot \bar{X})^{\mathrm{T}}(F - 1_N \cdot \bar{X})$$

And $1_N$ = A square matrix of ones or all-ones matrix.

**Code:**

```
% CSCM 70 ----------- LAB 05 -----------
% CSCM 70 ----------- 2154638 -----------
% ================================================================
% ================================================================
% PALLAV SHUKLA

% ------------ Covariance Matrix --------------------
% ---------------------------------------------------------------
% This function returns covariance of a matrix F in the variable res
function res = cov_manual(F)
% N denotes the size of the Matrix
N = size(F,1);
% finding the mean of data
Xmean = [sum(F,1)] / N;
%Calculating co-variance
covariance = ...
    (1/(N-1))*(transpose(F - ones(N) .* Xmean)*(F - ones(N) .* Xmean));
% Return value of the function
res = covariance;
% ====================================================================
%  ----------- END Covariance Matrix -----------
% ====================================================================
```

# OUTPUT MATLAB: COMMAND WINDOW

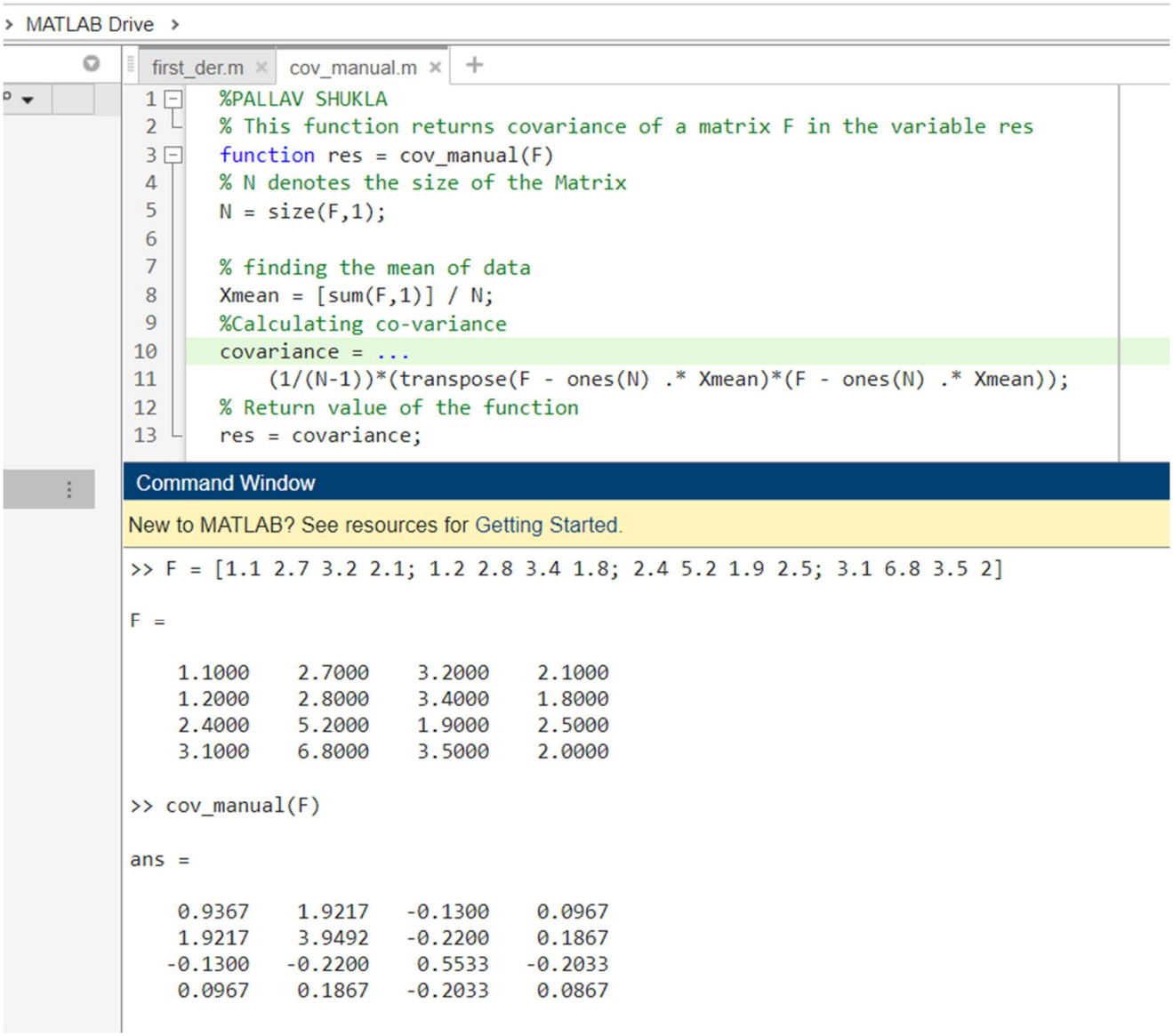



Image: 1

**Comparing/Verifying** with built-in function.

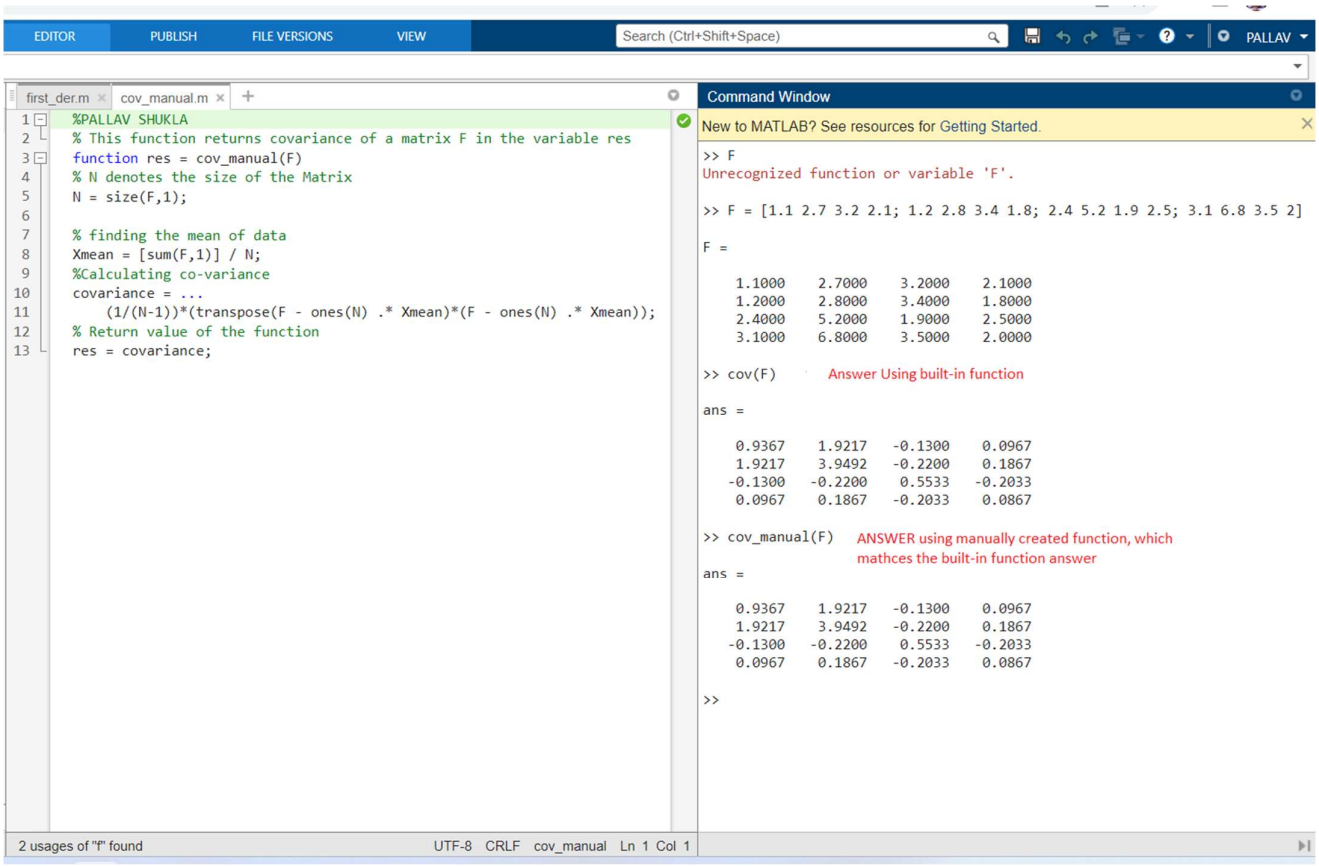



Image 2:

---

**Question 3: ->**

**Exercise 3** (Matlab). Start with data $x_i = y_i = i$ for $i$ ranging from 1 to 10. Add random perturbations of size up to 0.1 to each value. Compute the covariance matrix of the result, and obtain the eigenvectors and eigenvalues. Interpret. [Note: here you can use matlab commands for finding covariance ('cov') and eigenvalue/eigenvectors ('eig') ]- 2 marks

## Answer 3:

Generating the 10 random values using the formula:

➔ **CODE 1: From values - 0.1 to 0.1 as mentioned on either side**.

```
% CSCM 70 ----------- LAB 05 -----------

% CSCM 70 ----------- 2154638 -----------
% ================================================================
% ================================================================
% PALLAV SHUKLA

% ------------ Random Perturbations --------------------
% ----------------------------------------------------------------
% fIRST Created the Matrix
>> F = [1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7  7; 8 8; 9 9; 10 10;]
% Defined Xi and Yi where Xi and Yi have a difference of .1(either side) and i has a
% range of 1 to 10
```

```
>> Xi = (0.1 + 0.1)*rand(1,10) - 0.1
>> Yi = (0.1 + 0.1)*rand(1,10) - 0.1

% Reshaped Xi and Yi
>> Xi_reshape = reshape(Xi,size(Xi,2),1)
>> Yi_reshape = reshape(Yi,size(Yi,2),1)
>> new_mat = [Xi_reshape Yi_reshape]

% Adding with F
Ultimate_mat = new_mat + F

% finding cov
>> cov(Ultimate_mat)
>> Ultimate_mat_new = cov(Ultimate_mat)

% Finding eigen
>> [V,D] = eig(Ultimate_mat_new)

% ====================================================================
%  ----------- END Random Perturbations -----------
% ====================================================================
```

# OUTPUT MATLAB: COMMAND WINDOW

```
>> F = [1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7  7; 8 8; 9 9; 10 10;]

F =

     1     1
     2     2
     3     3
     4     4
     5     5
     6     6
     7     7
     8     8
     9     9
    10    10

>> Xi = (0.1 + 0.1)*rand(1,10) - 0.1

Xi =

  Columns 1 through 9

    0.0561   -0.0221   -0.0517   -0.0192   -0.0807   -0.0736    0.0884    0.0912    0.0150

  Column 10

   -0.0880
```

Image : 1

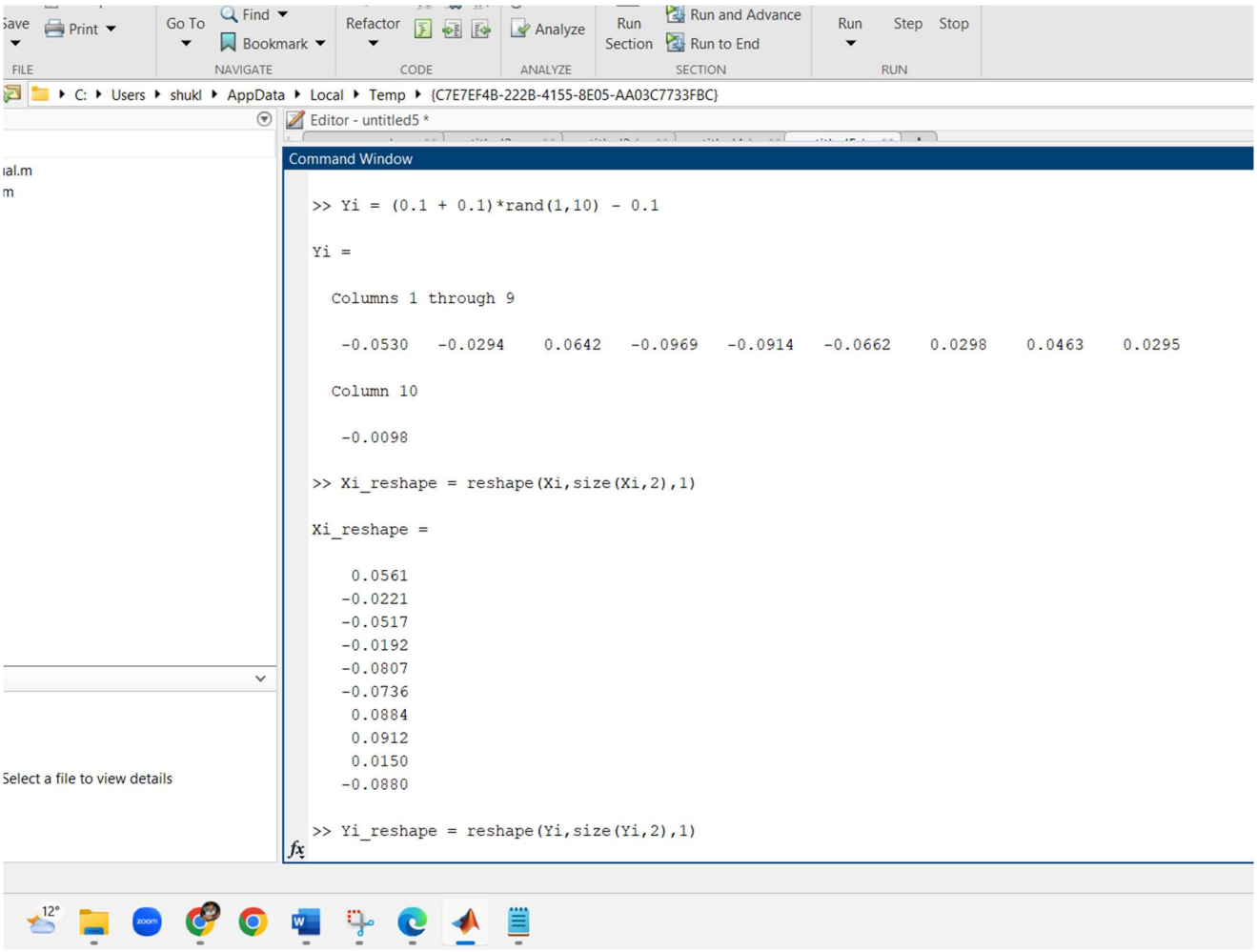

Save
Print
FILE
Go To
Find
Bookmark
NAVIGATE
Refactor
CODE
Analyze
ANALYZE
Run Section
Run and Advance
Run to End
SECTION
Run
Step
Stop
RUN
C: ▸ Users ▸ shukl ▸ AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
Command Window
>> Yi = (0.1 + 0.1)*rand(1,10) - 0.1
Yi =
Columns 1 through 9
-0.0530 -0.0294 0.0642 -0.0969 -0.0914 -0.0662 0.0298 0.0463 0.0295
Column 10
-0.0098
>> Xi_reshape = reshape(Xi,size(Xi,2),1)
Xi_reshape =
0.0561
-0.0221
-0.0517
-0.0192
-0.0807
-0.0736
0.0884
0.0912
0.0150
-0.0880
>> Yi_reshape = reshape(Yi,size(Yi,2),1)
Select a file to view details

Image : 2

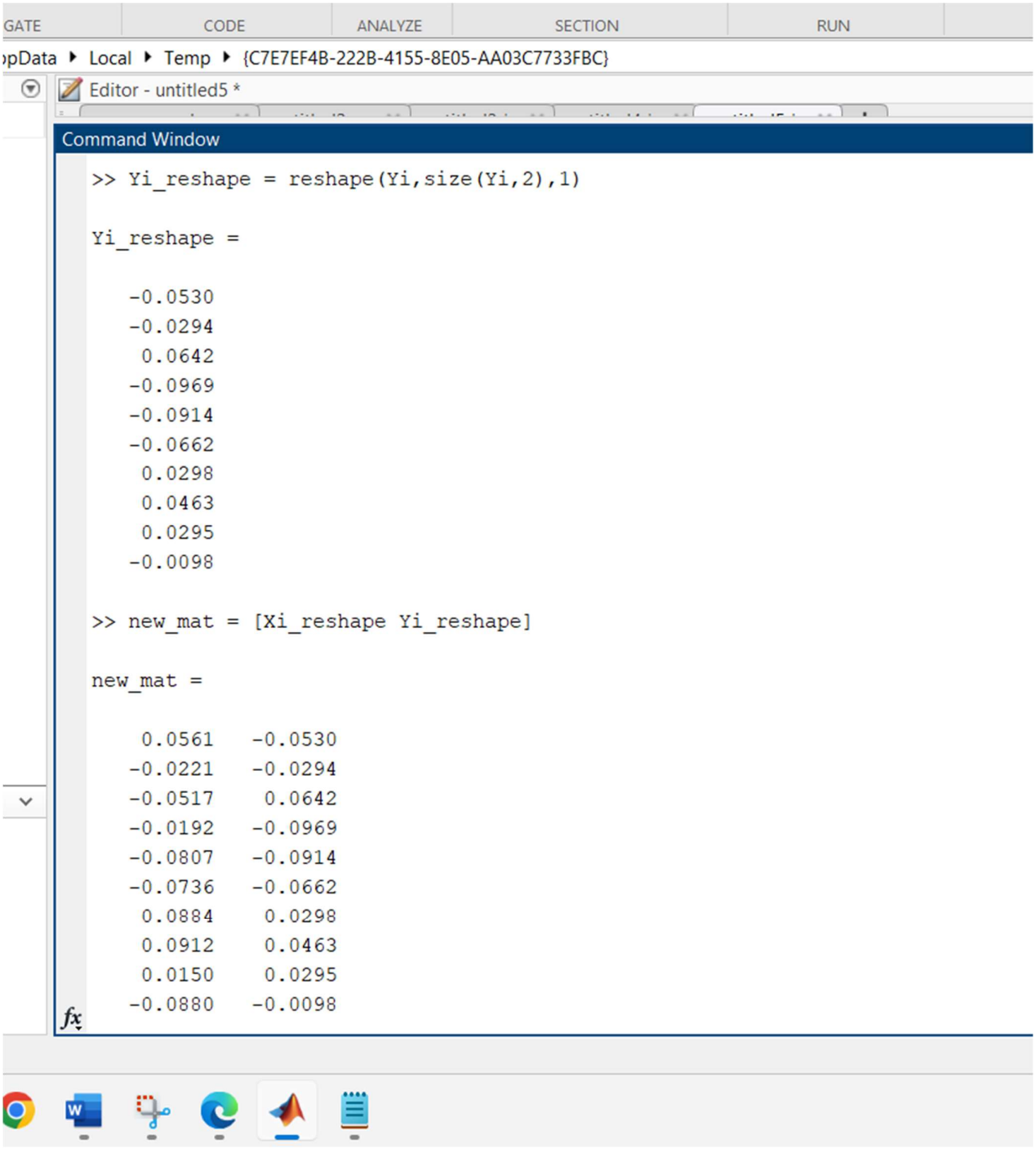

GATE
CODE
ANALYZE
SECTION
RUN
pData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
Command Window
>> Yi_reshape = reshape(Yi,size(Yi,2),1)
Yi_reshape =
-0.0530
-0.0294
0.0642
-0.0969
-0.0914
-0.0662
0.0298
0.0463
0.0295
-0.0098
>> new_mat = [Xi_reshape Yi_reshape]
new_mat =
0.0561 -0.0530
-0.0221 -0.0294
-0.0517 0.0642
-0.0192 -0.0969
-0.0807 -0.0914
-0.0736 -0.0662
0.0884 0.0298
0.0912 0.0463
0.0150 0.0295
-0.0880 -0.0098

Image : 3

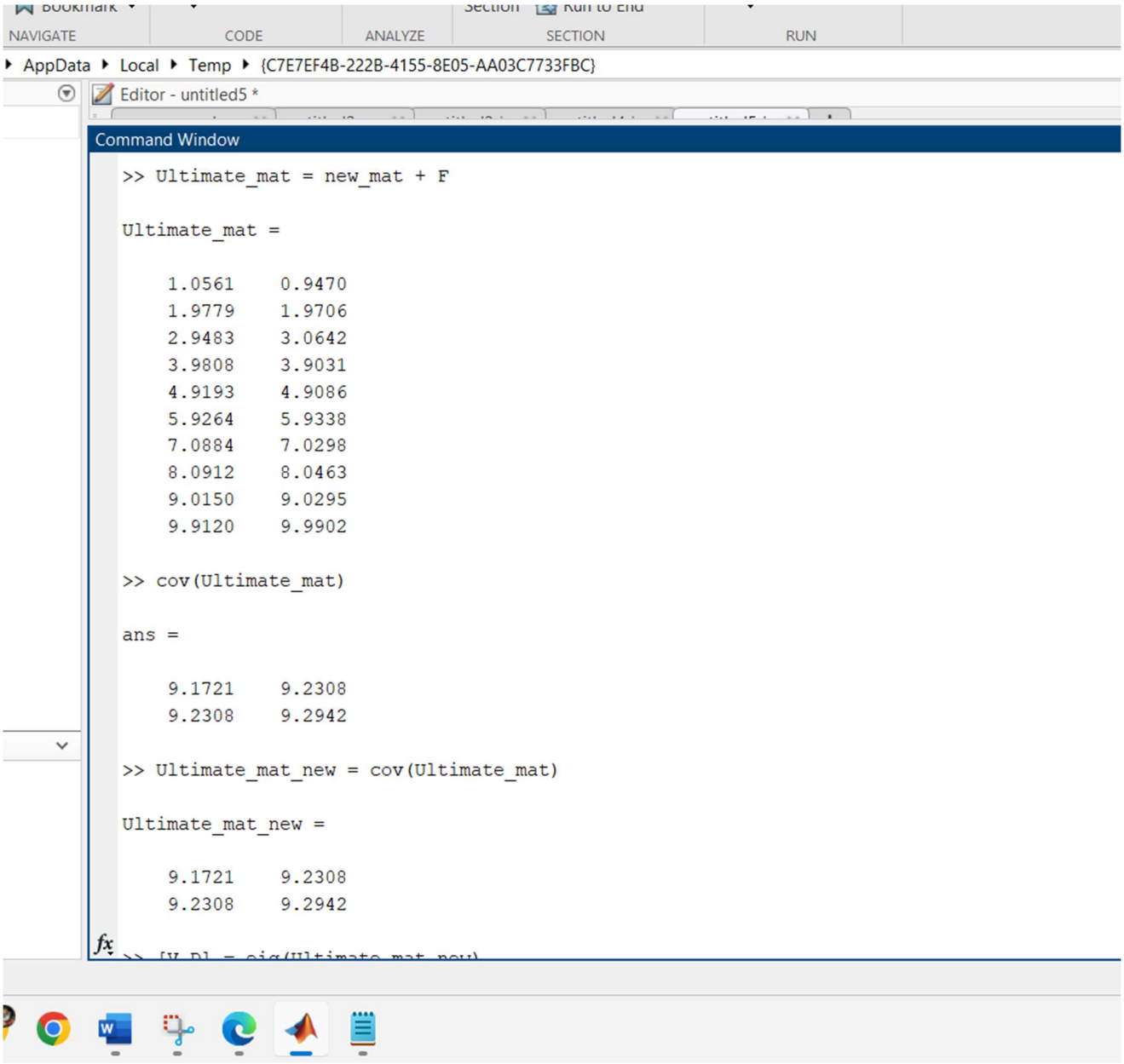

NAVIGATE
CODE
ANALYZE
SECTION
RUN
AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
Command Window
>> Ultimate_mat = new_mat + F
Ultimate_mat =
1.0561 0.9470
1.9779 1.9706
2.9483 3.0642
3.9808 3.9031
4.9193 4.9086
5.9264 5.9338
7.0884 7.0298
8.0912 8.0463
9.0150 9.0295
9.9120 9.9902
>> cov(Ultimate_mat)
ans =
9.1721 9.2308
9.2308 9.2942
>> Ultimate_mat_new = cov(Ultimate_mat)
Ultimate_mat_new =
9.1721 9.2308
9.2308 9.2942

Image : 4

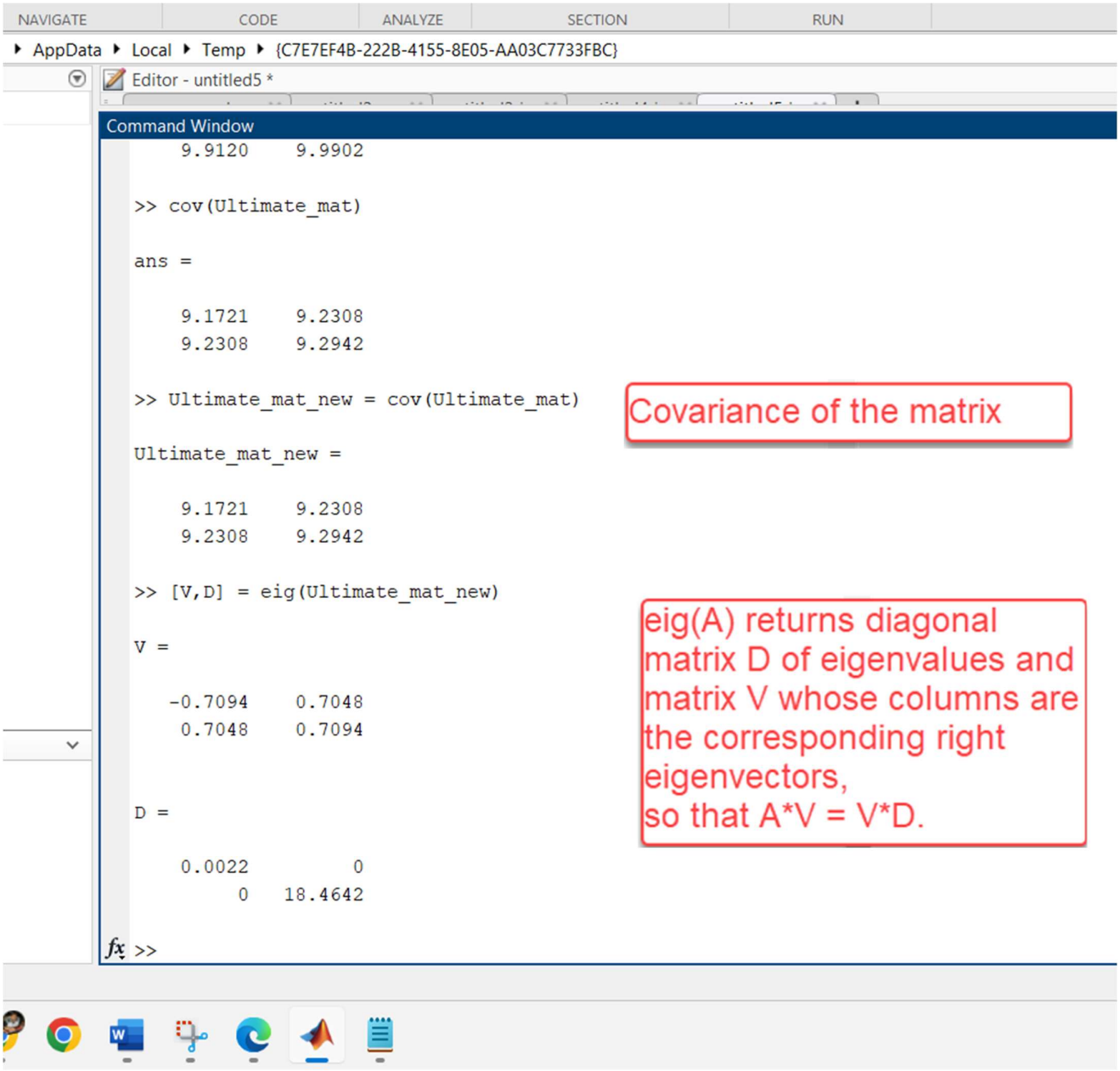



Image : 5

Covariance is a measure of the extent to which corresponding elements from two sets of ordered data move in the same direction[4]. An eigenvector of A is a vector that is taken to a multiple of itself by the matrix transformation T(x)=Ax, which explains the terminology[5]. Here we have a-lot of directions and we are going for only 2 directions. In the above code First we create the F matrix, then we define Xi and Yi, as per question with constraints of 0.1 and range between 1 to 10 and then we reshape it for better understanding and adding F to the newly formed reshaped matrix then completing the answer with finding co-variance and Eigen vector and value. In total when the value changes for question it doesn't make such a big difference and if there is a change it goes for all the values in that direction.

➔ CODE 2: trying for values 0 to 0.1

```
% CSCM 70 ----------- LAB 05 -----------
% CSCM 70 ----------- 2154638 -----------
% ================================================================
% ================================================================
% PALLAV SHUKLA

% ------------ Random Perturbations --------------------
% ----------------------------------------------------------------
% fIRST Created the Matrix
>> F = [1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9; 10 10;]

% Defined Xi and Yi where Xi and Yi have a difference of .1 and i has a
% range of 1 to 10
>> Xi = 0.1*rand(1,10)
>> Yi = 0.1*rand(1,10)

% Reshaped Xi and Yi
>> Xi_reshape = reshape(Xi,size(Xi,2),1)
>> Yi_reshape = reshape(Yi,size(Yi,2),1)
>> new_mat = [Xi_reshape Yi_reshape]

% Adding up with F
Ultimate_mat = new_mat + F

% finding cov
>> cov(Ultimate_mat)
>> Ultimate_mat_new = cov(Ultimate_mat)

% Finding eigen
>> [V,D] = eig(Ultimate_mat_new)

% ================================================================
%  ----------- END Random Perturbations -----------
% ================================================================
```

# OUTPUT MATLAB: COMMAND WINDOW

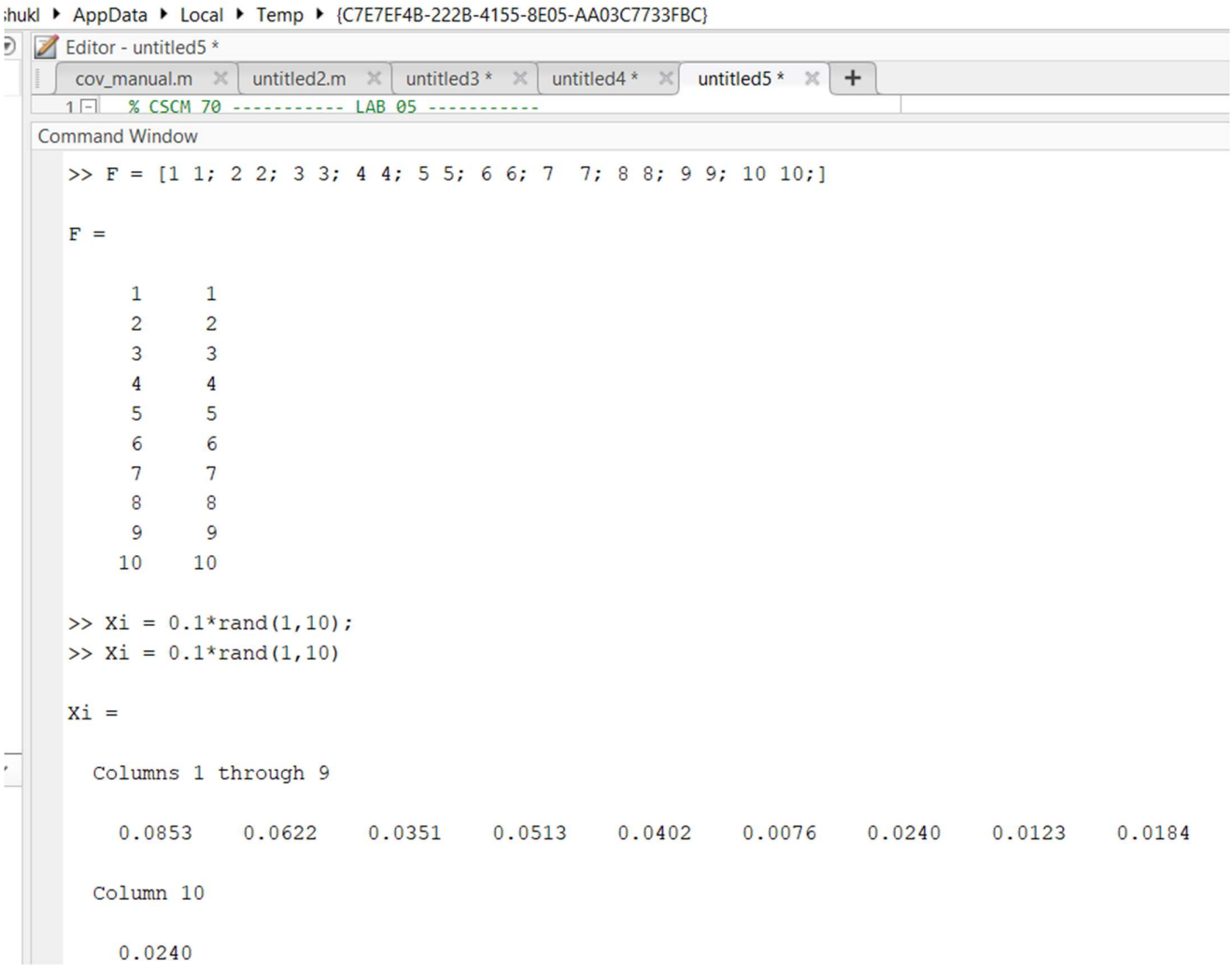

shukl ▸ AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
cov_manual.m
untitled2.m
untitled3 *
untitled4 *
untitled5 *
1 % CSCM 70 ---------- LAB 05 ----------
Command Window
>> F = [1 1; 2 2; 3 3; 4 4; 5 5; 6 6; 7 7; 8 8; 9 9; 10 10;]
F =
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
>> Xi = 0.1*rand(1,10);
>> Xi = 0.1*rand(1,10)
Xi =
Columns 1 through 9
0.0853 0.0622 0.0351 0.0513 0.0402 0.0076 0.0240 0.0123 0.0184
Column 10
0.0240

Image : 1

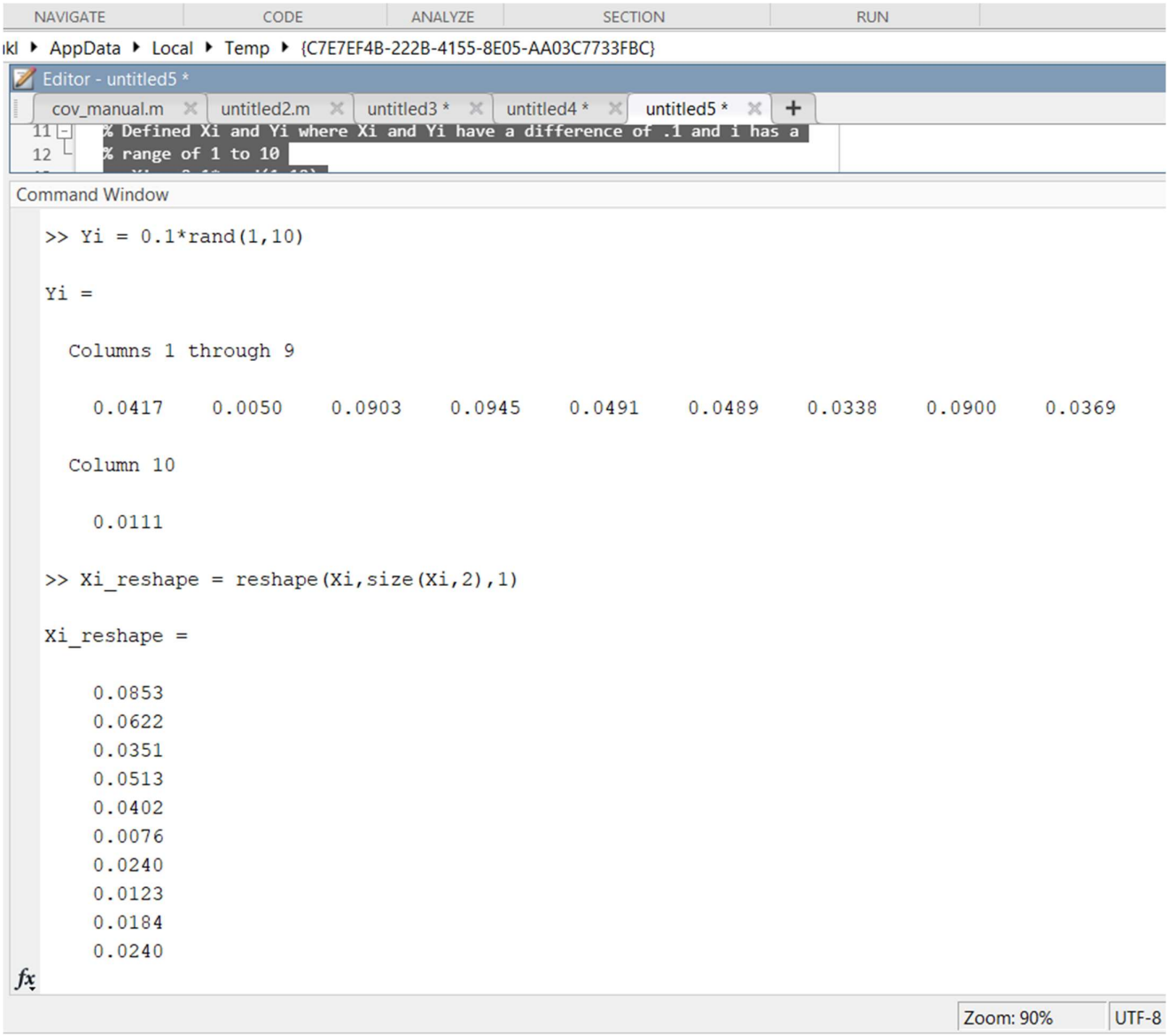

NAVIGATE
CODE
ANALYZE
SECTION
RUN
ikl ▸ AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
cov_manual.m
untitled2.m
untitled3 *
untitled4 *
untitled5 *
11 % Defined Xi and Yi where Xi and Yi have a difference of .1 and i has a
12 % range of 1 to 10
Command Window
>> Yi = 0.1*rand(1,10)
Yi =
Columns 1 through 9
0.0417 0.0050 0.0903 0.0945 0.0491 0.0489 0.0338 0.0900 0.0369
Column 10
0.0111
>> Xi_reshape = reshape(Xi,size(Xi,2),1)
Xi_reshape =
0.0853
0.0622
0.0351
0.0513
0.0402
0.0076
0.0240
0.0123
0.0184
0.0240
Zoom: 90%
UTF-8

Image: 2

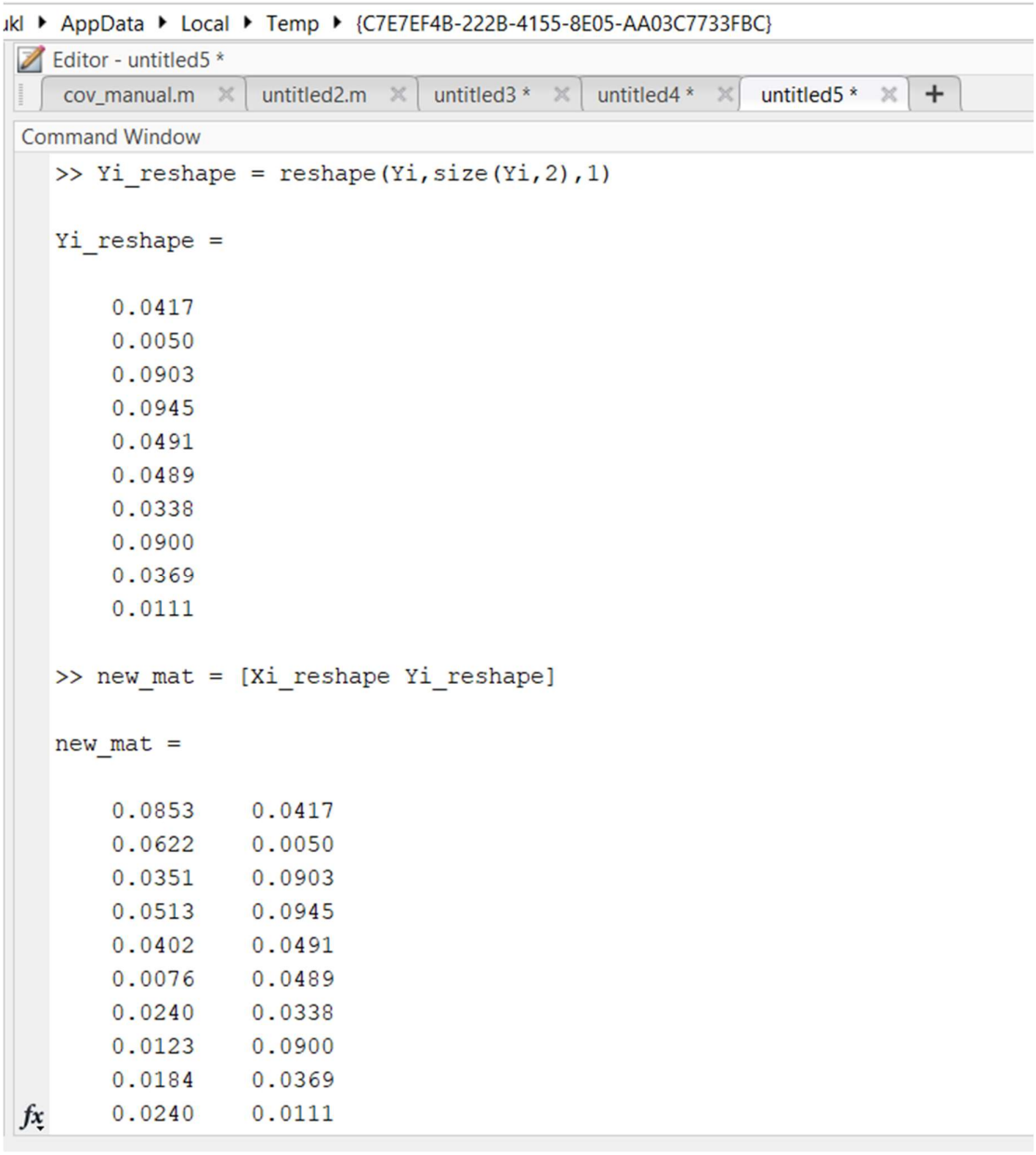

ıkl ▸ AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
cov_manual.m
untitled2.m
untitled3 *
untitled4 *
untitled5 *
Command Window
>> Yi_reshape = reshape(Yi,size(Yi,2),1)
Yi_reshape =
0.0417
0.0050
0.0903
0.0945
0.0491
0.0489
0.0338
0.0900
0.0369
0.0111
>> new_mat = [Xi_reshape Yi_reshape]
new_mat =
0.0853 0.0417
0.0622 0.0050
0.0351 0.0903
0.0513 0.0945
0.0402 0.0491
0.0076 0.0489
0.0240 0.0338
0.0123 0.0900
0.0184 0.0369
0.0240 0.0111

Image: 3

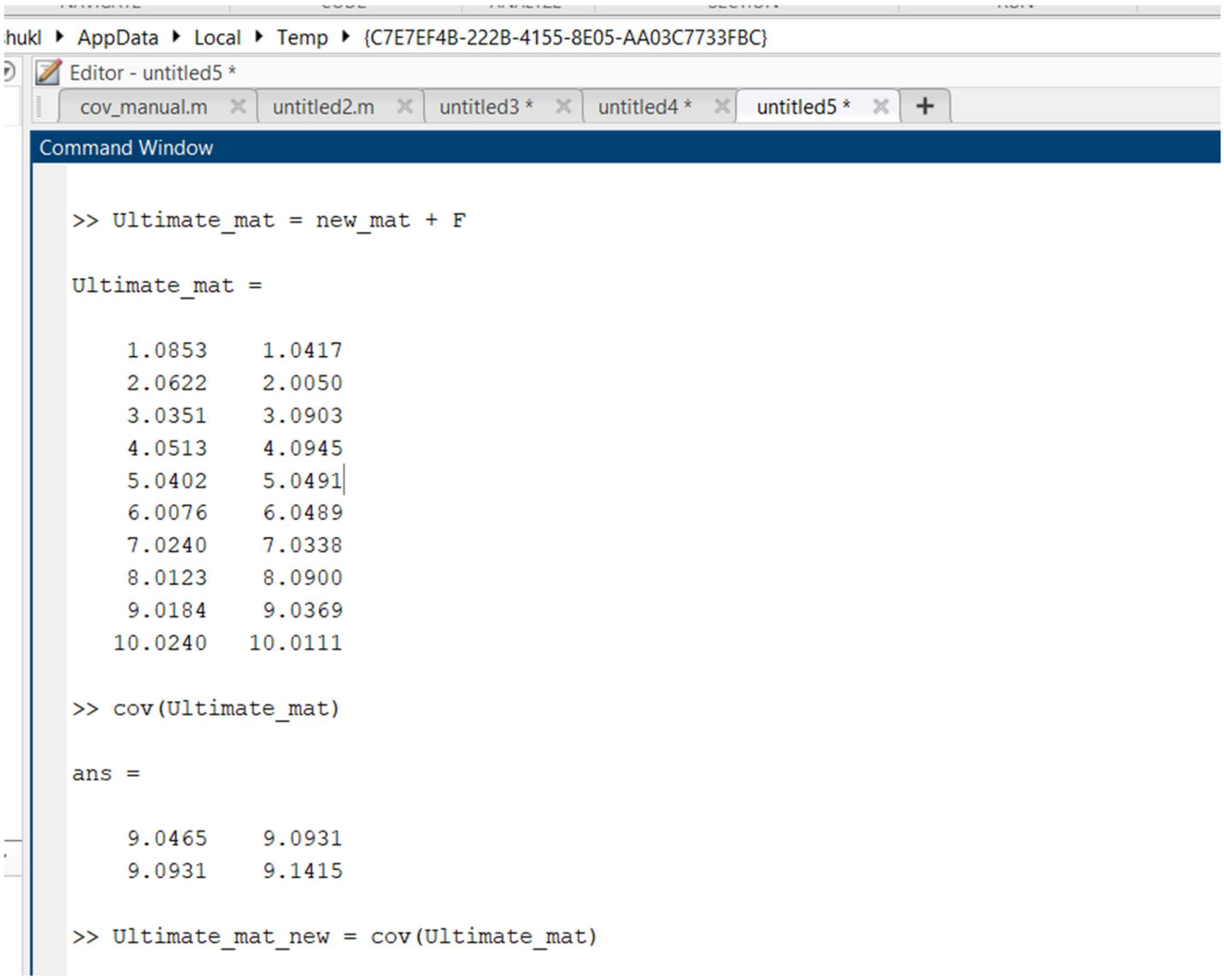

hukl ▸ AppData ▸ Local ▸ Temp ▸ {C7E7EF4B-222B-4155-8E05-AA03C7733FBC}
Editor - untitled5 *
cov_manual.m
untitled2.m
untitled3 *
untitled4 *
untitled5 *
Command Window
>> Ultimate_mat = new_mat + F
Ultimate_mat =
1.0853 1.0417
2.0622 2.0050
3.0351 3.0903
4.0513 4.0945
5.0402 5.0491
6.0076 6.0489
7.0240 7.0338
8.0123 8.0900
9.0184 9.0369
10.0240 10.0111
>> cov(Ultimate_mat)
ans =
9.0465 9.0931
9.0931 9.1415
>> Ultimate_mat_new = cov(Ultimate_mat)

Image: 4

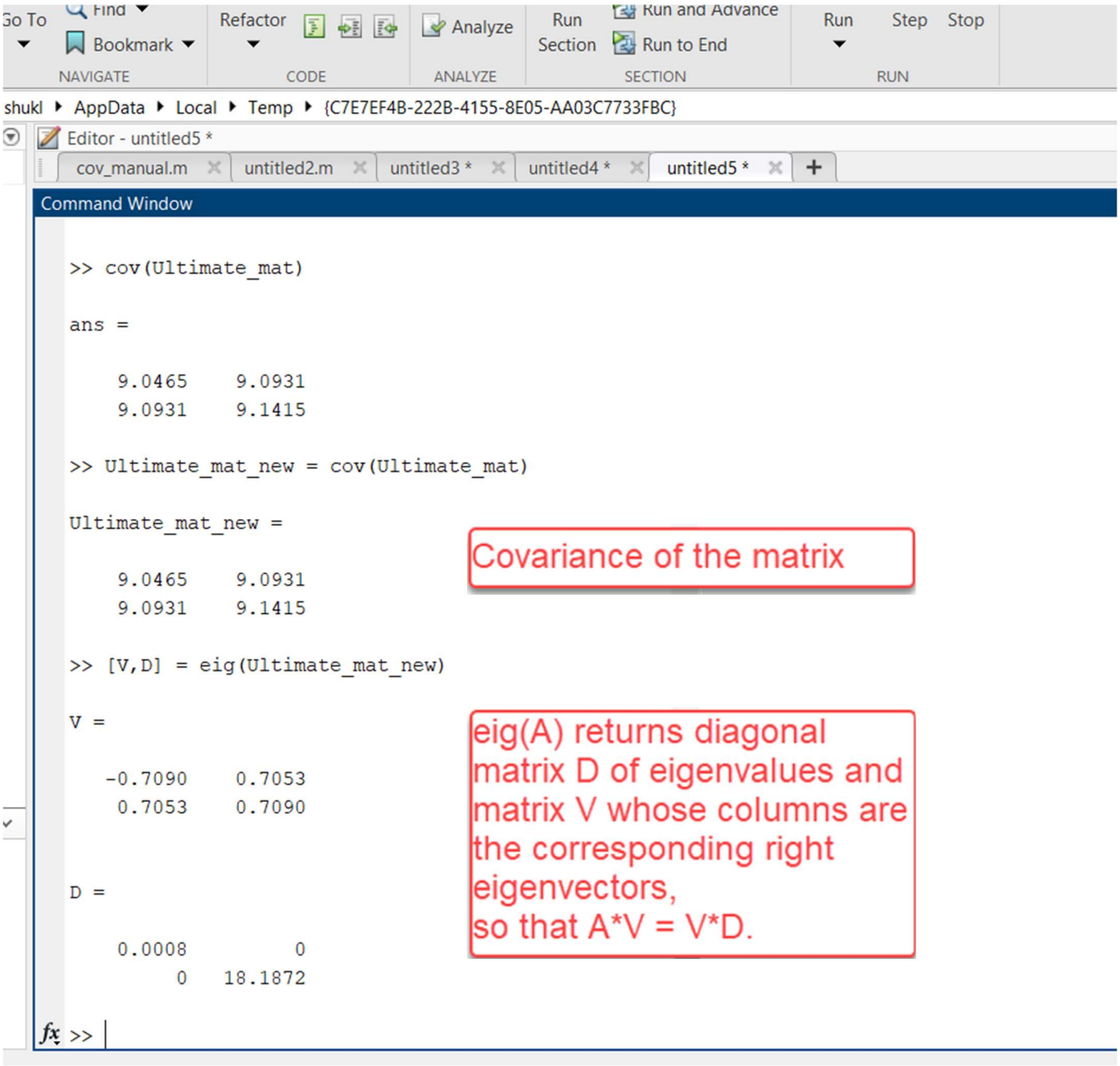
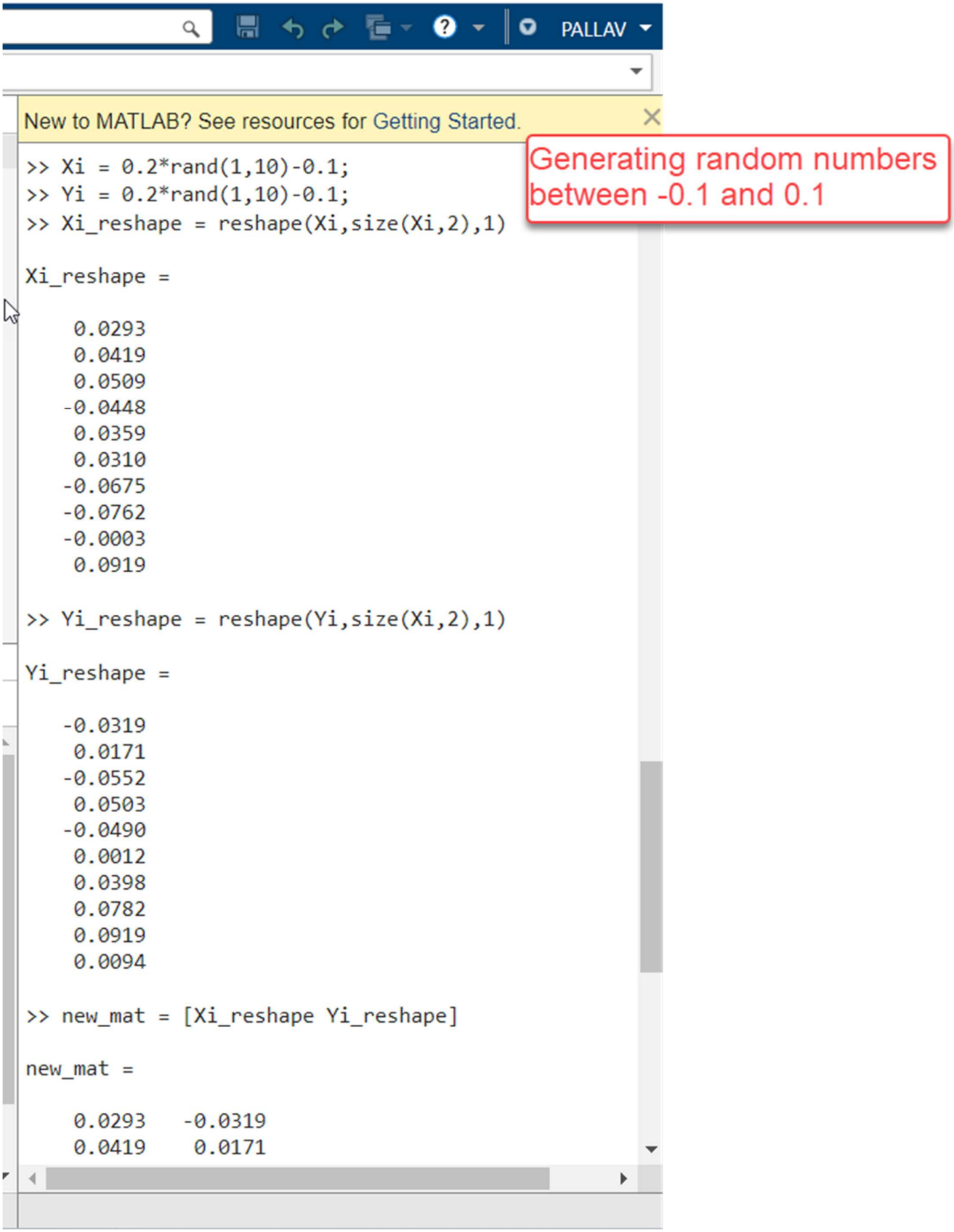
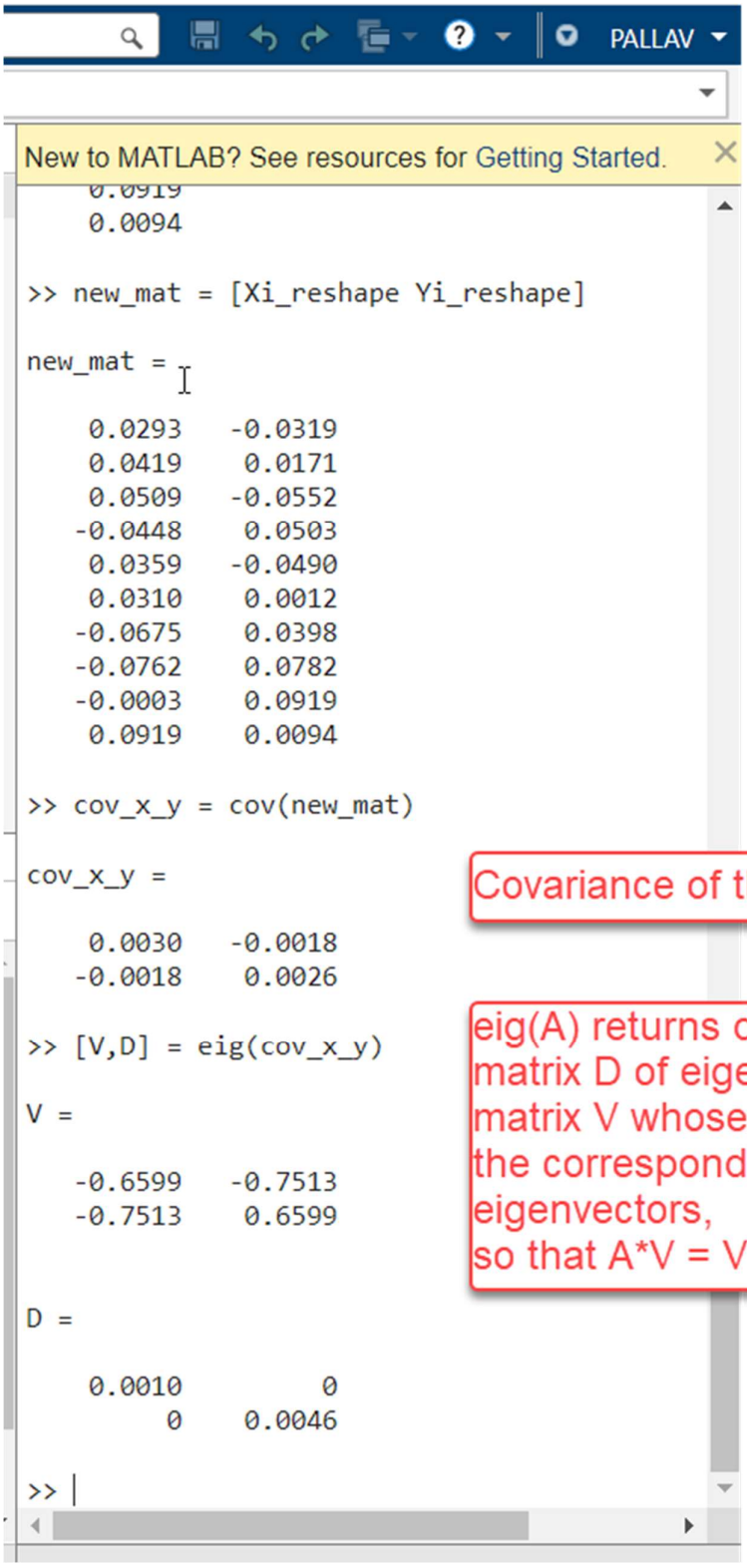



Image: 5

➔ Tried for the range of -0.1 to 0.1 without genrating F-> Next page



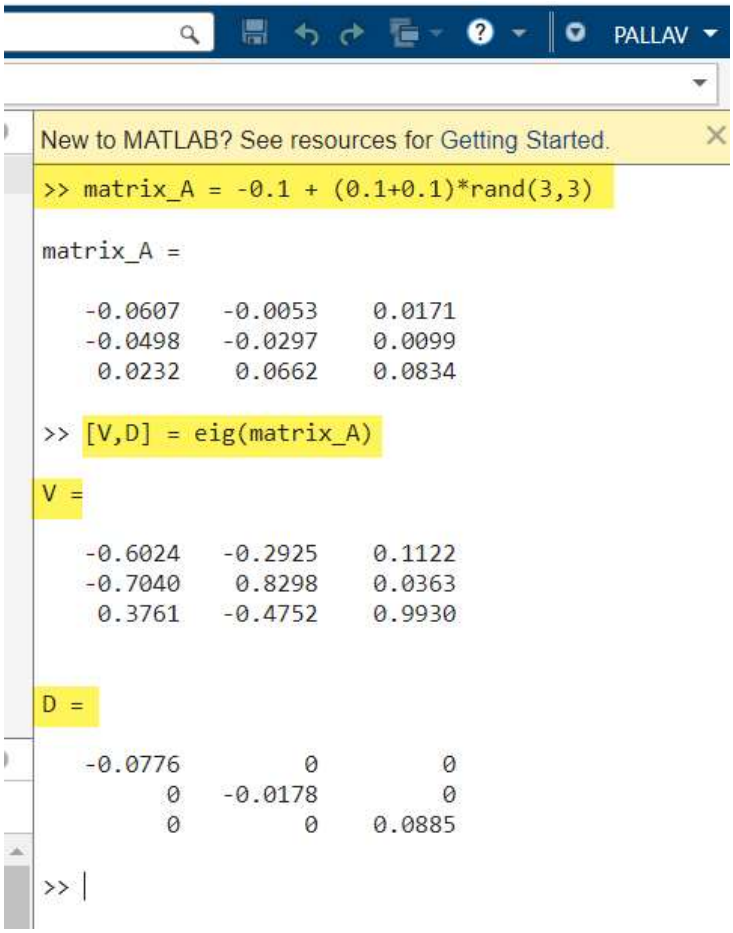

Then finding the eigen values and eigen vectors




Covariance of the matrix

eig(A) returns diagonal matrix D of eigenvalues and matrix V whose columns are the corresponding right eigenvectors, so that A*V = V*D.

➔ Trying eigen function on randomly generated (3,3) matrix:




---

---

## References:


1.) Sir's Lecture.
- Panopto. Panopto. Retrieved November 10, 2022, from https://swanseauniversity.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=164ff299-6a2c-4aa2-af57-af4600c7f45c

2.) Canvas Panopto. Panopto. Retrieved November 10, 2022, from https://swanseauniversity.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=164ff299-6a2c-4aa2-af57-af4600c7f45c

3.) Intercept to zero .
*When should we force the intercept to zero? | Researchgate*. Retrieved November 14, 2022, from https://www.researchgate.net/post/When-should-we-force-the-intercept-to-zero

4.) Covariance
Berman H.B., "*Variance-Covariance Matrix*", [online] Available at: https://stattrek.com/matrix-algebra/covariance-matrix URL [Accessed Date: 11/14/2022].

5.) EigenVector
Dan Margalit, J. R. *Interactive linear algebra*. Eigenvalues and Eigenvectors. Retrieved November 14, 2022, from https://textbooks.math.gatech.edu/ila/eigenvectors.html

## End

++++++++++++++++++ Lab 005 ++++++++++++++++++

++++++++++++++++++ 2154638 ++++++++++++++++++