

Rita Herfi
Professor Nguyen
EE 300
15 Aug 2020

EE 300 Matlab Project Report

%Variables, Functions and full code can be found starting on page 6.

%I would first like to begin by saying that this is my first time using matlab and I apologize in advance if the code at the bottom is a bit messy.

- 1) i) Use MATLAB functions mean(x) and mean(y).

Result:

mx = 25.1184

my = 6.5289

-
- ii) write your own function my_mean(x).

* can be found in FUNCTIONS area.

Result:

my_mean(x) = 25.1184

my_mean(y) = 6.5289

%As you will see in the script later, I had a lot of trial and errors with the functions but eventually got to where I needed to be, yielding the same results as the functions previously implemented in matlab.

2) i) Find sample std using std(x) and std(y).

Result:

stdx = 8.0237

stdy = 0.7763

ii) write your own function my_std(x), must include my_mean function.

* can be found in FUNCTIONS area.

Results:

my_std(x) = 8.0237

my_std(y) = 0.7763

%I had a bit more trouble with this function since I couldn't decide how to fix the loop and some variables but I eventually played around with what I found suitable with placing specific parentheses. I also made sure to use ii in the loop since the guide suggested to do so, since we don't want to confuse i with imaginary numbers.

- 3) i) use matlab function median(x).

Result:

medianx = 24.9000

- ii) create your own function, my_median.

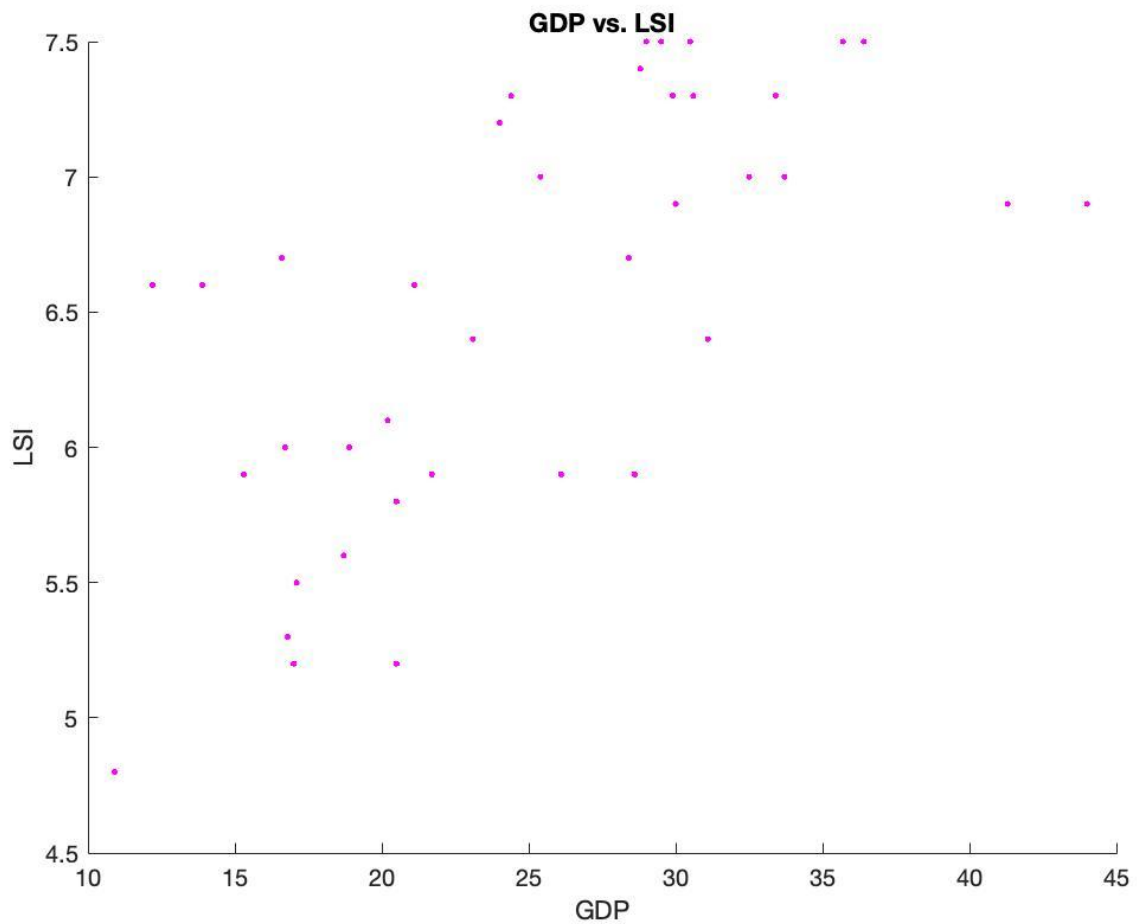
* can be found in FUNCTIONS area.

Result:

my_median(x) = 24.9000

%Although median is a simple function I struggled with placing the correct function in the correct areas. I later also remembered that for even numbers there is a specific formula to get the true median and that was why I kept getting the incorrect answer. Once I implemented the correct function however, the answers matched and I tested median(y) and my_median(y) just in case to see if it would match and they did.

4) Try scatter(x,y).



%Since the scale only contains positive numbers we can see that this plot resides in the first quadrant of a graph, the positive quadrant, and therefore we can conclude that it has a positive correlation.

- 5) Find sample covariance.

*command can be found in COMMAND WINDOW FULL CODE section.

Result:

$$\text{covxy} = 4.1738$$

- 6) Find correlation coefficient using sample covariance and standard deviation.

*command can be found in COMMAND WINDOW FULL CODE section.

Result:

$$r = 0.6701$$

%The calculation confirms with the observation from the scatter plot since our correlation coefficient is not 0 and is a positive number, thus concluding and proving that the correlation is positive.

VARIABLES:

Workspace	
Name ▲	Value
ans	6.6500
covxy	4.1738
medianx	24.9000
mediany	6.6500
mx	25.1184
my	6.5289
N	38
r	0.6701
stdx	8.0237
stdy	0.7763
x	1x38 double
y	1x38 double

FUNCTIONS:

Variables.m

%Variable Definitions

```
x = [33.4 32.5 30.0 29.9 16.6 21.1 29.0 18.7 29.5 31.1 33.7 17.0 16.8 30.5 25.4...
      24.0 26.1 28.6 21.7 15.3 41.3 13.9 28.8 24.4 35.7 18.9 20.5 20.2 20.5 23.1 ...
      30.6 36.4 17.1 28.4 44.0 12.2 16.7 10.9]
y = [7.3 7.0 6.9 7.3 6.7 6.6 7.5 5.6 7.5 6.4 7.0 5.2 5.3 7.5 7.0 7.2 5.9 5.9 ...
      5.9 5.9 6.9 6.6 7.4 7.3 7.5 6.0 5.2 6.1 5.8 6.4 7.3 7.5 5.5 6.7 6.9 6.6 6 ...
      4.8]
```

my_mean.m

% Creates my_mean variable

function my_mean = my_mean(x)

%initializes variable

my_mean = 0;

%n is equal to the number of elements in the array

n = numel(x);

%sum of the elements in the array

sum_var = sum (x);

```
%actual equation for average
my_mean = sum_var/n;
end
```

my_std.m

```
%Creates standard deviation function
function my_std = my_std(x)
%initialize std
my_std = 0;
m = my_mean(x);
%make a for loop to subtract all elements
for ii=1:numel(x)
my_std=my_std+((x(ii)-m)^2);
end
%k is equal to 1 divided by the number of elements in the array - 1
k = 1/(numel(x)-1);
v = k*my_std;
my_std = sqrt(v);
end
```

my_median.m

```
%Create median
function my_median = my_median(x)
%sorts data into xa
xa = sort(x);
%number of elements in sorted array, (should be same as input array)
n = numel(xa);
%i found rem in the catalog, uses same style for even and odd as C++
if rem(n,2) ~= 0 %if n is odd at 1 to make it even
    n = (n+1)/2;
    my_median = xa(n);
%if n is even then add the two middle elements and divide by 2
else
    n = n/2;
    my_median = (xa(n)+xa(n+1))/2;
end
end
```

COMMAND WINDOW FULL CODE:

```
>> variables
```

```
x =
```

```
Columns 1 through 5
```

```
33.4000 32.5000 30.0000 29.9000 16.6000
```

```
Columns 6 through 10
```

```
21.1000 29.0000 18.7000 29.5000 31.1000
```

```
Columns 11 through 15
```

```
33.7000 17.0000 16.8000 30.5000 25.4000
```

```
Columns 16 through 20
```

```
24.0000 26.1000 28.6000 21.7000 15.3000
```

```
Columns 21 through 25
```

```
41.3000 13.9000 28.8000 24.4000 35.7000
```

```
Columns 26 through 30
```

```
18.9000 20.5000 20.2000 20.5000 23.1000
```

```
Columns 31 through 35
```

```
30.6000 36.4000 17.1000 28.4000 44.0000
```

```
Columns 36 through 38
```

```
12.2000 16.7000 10.9000
```


y =

Columns 1 through 5

7.3000 7.0000 6.9000 7.3000 6.7000

Columns 6 through 10

6.6000 7.5000 5.6000 7.5000 6.4000

Columns 11 through 15

7.0000 5.2000 5.3000 7.5000 7.0000

Columns 16 through 20

7.2000 5.9000 5.9000 5.9000 5.9000

Columns 21 through 25

6.9000 6.6000 7.4000 7.3000 7.5000

Columns 26 through 30

6.0000 5.2000 6.1000 5.8000 6.4000

Columns 31 through 35

7.3000 7.5000 5.5000 6.7000 6.9000

Columns 36 through 38

6.6000 6.0000 4.8000

```
>> save('data.mat','x','y')
```

```
>> load('data.mat')
```

```
>> mean(x);
```

```
>> mx = mean(x);
```

```
>> my = mean(y);
```

```
>> my_mean(x)
Unrecognized function or variable 'my_mean'.
```

```
>> my_mean(x)
Error: File: my_mean.m Line: 1 Column: 19
Invalid expression. Check for missing or extra
characters.
```

```
>> my_mean(x)
Error: File: my_mean.m Line: 5 Column: 8
Invalid expression. When calling a function or
indexing a variable, use parentheses. Otherwise,
check for mismatched delimiters.
```

```
>> my_mean(x)
Error: File: my_mean.m Line: 5 Column: 8
Invalid expression. When calling a function or
indexing a variable, use parentheses. Otherwise,
check for mismatched delimiters.
```

```
>> my_mean(x)
Error: File: my_mean.m Line: 5 Column: 8
Invalid expression. When calling a function or
indexing a variable, use parentheses. Otherwise,
check for mismatched delimiters.
```

```
>> my_mean(x)
Error: File: my_mean.m Line: 6 Column: 8
Invalid expression. Check for missing multiplication
operator, missing or unbalanced delimiters, or other
syntax error. To construct matrices, use brackets
instead of parentheses.
```

```
>> my_mean(x)
Error using /
Matrix dimensions must agree.
```

```
Error in my_mean (line 5)
my_mean = (sum_var/n);
```

```
>> my_mean(x)
```

```
ans =
```

```
25.1184
```

```
>> my_mean(y)
```

```
ans =
```

```
6.5289
```

```
>> my_mean(x)
```

```
ans =
```

```
25.1184
```

```
>> my_mean(x)
```

```
ans =
```

```
25.1184
```

```
>> my_mean(y)
```

```
ans =
```

```
6.5289
```

```
>> my_mean(y)
```

```
ans =
```

```
6.5289
```

```
>>
```

```
>> edit
```

```
>> stdx = std(x);
```

```
>> stdy = std(y);
```

```
>> my_std(x)
```

Error: File: my_std.m Line: 10 Column: 1

Using identifier 'my_std' as both a variable and a command is not supported. For more information, see "How MATLAB Recognizes Command Syntax".

```
>> my_std(x)
```

Array indices must be positive integers or logical values.

Error in my_std (line 10)

```
my_std = my_std+(x(i)-my_mean(x));
```

```
>> my_std(x)
```

Array indices must be positive integers or logical values.

Error in my_std (line 10)

```
my_std=my_std+(x(i)-my_mean(x));
```

```
>> my_std(x)
```

```
ans =
```

```
1.9566e-14
```

```
>> my_std(x)
```

```
ans =
```

```
1.9566e-14
```

```
>> my_std(x)
```

Error: File: my_std.m Line: 8 Column: 36

Invalid expression. When calling a function or indexing a variable, use parentheses. Otherwise,

check for mismatched delimiters.

```
>> my_std(x)
```

```
ans =
```

```
0.0000 +24.9437i
```

```
>> my_std(x)
```

```
ans =
```

```
0.0000 +24.9437i
```

```
>> my_std(x)
```

```
ans =
```

```
0.0000 +24.9437i
```

```
>> my_std(x)
```

```
ans =
```

```
8.0237
```

```
>> my_std(y)
```

```
ans =
```

```
0.7763
```

```
>> edit
```

```
>> medianx=median(x);
```

```
>> my_median(x);
```

```
>> my_median(x)
```

```
ans =
```

19

```
>> my_median(x)
```

```
ans =
```

```
24.4000
```

```
>> my_median(x)
```

```
Unrecognized function or variable 'odd'.
```

```
Error in my_median (line 15)
```

```
if n == odd
```

```
>> my_median(x)
```

```
Index exceeds the number of array elements  
(38).
```

```
Error in my_median (line 23)
```

```
my_median = (xa(n)+xa(p))/2;
```

```
>> my_median(x)
```

```
Index exceeds the number of array elements  
(38).
```

```
Error in my_median (line 23)
```

```
my_median = (xa(n)+xa(n+1))/2;
```

```
>> my_median(x)
```

```
>> my_median(x)
```

```
>> my_median(x)
```

```
Unrecognized function or variable 'odd'.
```

```
Error in my_median (line 16)
```

```
if n == odd
```

```
>> my_median(x)
```

```
Index exceeds the number of array elements  
(38).
```

Error in my_median (line 20)

```
my_median = (xa(n)+xa(n+1))/2;
```

```
>> my_median(x)
```

```
ans =
```

```
24.4000
```

```
>> my_median(x)
```

```
ans =
```

```
24.4000
```

```
>> my_median(x)
```

```
ans =
```

```
24.9000
```

```
>> my_median(x)
```

```
Index exceeds the number of array elements  
(38).
```

Error in my_median (line 15)

```
my_median = (xa(n)+xa(p))/2;
```

```
>> my_median(x)
```

```
ans =
```

```
24.9000
```

```
>> my_median(y)
```

```
ans =
```

6.6500

```
>> mediany = median(y)
```

```
mediany =
```

6.6500

```
>> scatter(x,y,'g')
```

```
>> scatter(x,y,)
```

```
scatter(x,y,)
```

↑

Error: Invalid expression. When calling a function or indexing a variable, use parentheses. Otherwise, check for mismatched delimiters.

```
>> scatter(x,y)
```

```
>> scatter(x,y,'r r')
```

Error using scatter (line 44)

Error in color/linetype argument.

```
>> scatter(x,y,' r')
```

```
>> scatter(x,y,' g')
```

```
>> scatter(x,y,' m')
```

```
>> xlabel('GDP')
```

```
>> ylabel('LSI')
```

```
>> title('GDP vs. LSI')
```

```
>> edit
```

```
>> samp_cov(x,y)
```

Unrecognized function or variable 'mx'.

Error in samp_cov (line 3)

```
samp_cov = (N/(N-1))*mean((x-mx)*(y-my));
```

```
>> N = numel(x);
```

```
>> covxy = (N/(N-1))*mean((x-mx)*(y-my))
```

Error using *

Incorrect dimensions for matrix

multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To perform elementwise multiplication, use '.*'.

```
>> covxy = (N/(N-1))*mean((x-mx).*(y-my))
```

```
covxy =
```

```
4.1738
```

```
>> sx=(x,0);
```

```
sx=(x,0);
```

```
↑
```

Error: Invalid expression. When calling a function or indexing a variable, use parentheses. Otherwise, check for mismatched delimiters.

```
>> sx=std(x,0);
```

```
>> sy=std(y,0);
```

```
>> r=covxy/(sx*xy)
```

Unrecognized function or variable 'xy'.

```
>> r=covxy/(sx*sy)
```

```
r =
```

```
0.6701
```

```
>> r=covxy/(stdx*stdy)
```

```
r =
```

```
0.6701
```

```
>>
```