CE300
MATLAB PROGRAMMING
DATA ANALYSIS

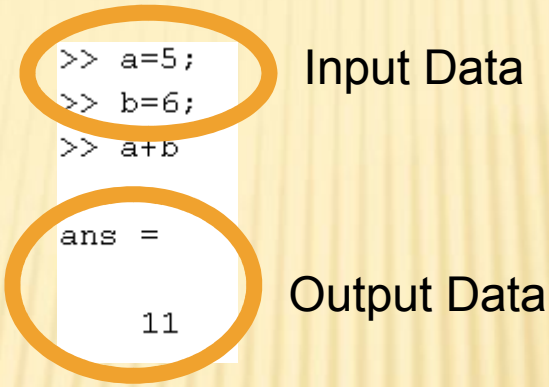
By Şevki ÖZTÜRK

OBJECTIVES

- INTRODUCTION TO DATA ANALYSIS
- READ/WRITE DATA
- STATISTICAL ANALYSIS
- DATA VISUALIZATION

INTRODUCTION TO DATA ANALYSIS

- Up to now, keyboard has been used for data input.



- If the amount of data is large and the same data will be used repeatedly, it is difficult to use keyboard.
- Another way of giving input data is to use a file having stored data. Matlab can read data, operate on data and write data.

INTRODUCTION TO DATA ANALYSIS

- With data analysis some properties and characteristics of data can be determined,

READ/WRITE DATA

File Opening/Closing

fileid=fopen(*filename, permission):* open the file according to the permission

Permission	Explanation
'r'	Opens the file to read (if the file exists)
'r+'	Opens the file to read and write (if the file exists)
'w'	Opens or creates a file to write. All existing data is deleted.
'w+'	Opens or creates a file to read and write. All existing data is deleted.
'a'	Open or create a file to write. The data stored before is not deleted
'a+'	Open or create a file to read and write. The before is not deleted

READ/WRITE DATA

File Opening/Closing

fclose *(fileid):* closes the file

fclose ('all') or fclose all: closes all files

```
fid=fopen('file1.txt','r')
```

```
fid=fopen('file1.txt','w')
```

```
fclose(fid)
```

READ/WRITE DATA

Writing information to a file

fprintf *(fileID,format, A):* applies format to all elements of A and writes data.

```
>> fid=fopen('file1.txt','w');
>> fprintf(fid,'%25s\n','Writing Data');
>> fprintf(fid,'Writing Data\n');
>> close all
```

Writes that

format

```
>> fid=fopen('file1.txt','w');
>> fprintf(fid,'Introduction to Matlab\n');
>> fprintf(fid,'Data Analysis\n');
```

READ/WRITE DATA

```
no,    team,     point,   positior
 1,     fb ,      82,       1
 2,     gs ,      46,       8
 3,     bjk,      54,       5
 4,     ts ,      82,       2
```

```
>> no=[1;2;3;4];
>> team=['fb ';'gs ';'bjk';'ts '];
>> points=[82;46;54;82];
>> position=[1;8;5;2];
>> teams=fopen('super_lig.txt','w');
>> fprintf(teams,'%7s,%7s,%9s,%10s\n','no','teams','point','position');
>> for i=1:length(no)
fprintf(teams,'%7d,%7s,%7d,%7d\n',...
no(i,:),team(i,:),points(i,:),position(i,:));
end
```

READ/WRITE DATA

Reading information from a file

fscanf *(fileID,format,size):* reads "size" elements according to the format given.

```
>> read=fopen('file1.txt','r');
>> fscanf(read,'%5c',1)

ans =

Intro
```

Reads 1 element

Reads 5 character (c) including space

READ/WRITE DATA

Saving data

save *filename*

```
>> x=[1,2,3,4];
>> y=[11,12,13,14];
>> save ex1
```

```
save ex2.txt
```

```
save -'ascii' ex2.txt
```

READ/WRITE DATA

Loading the data

load *filename*

```
>> load example.dat
>> example

example =

    10    20    30    40
     5    10    15    20
    30    40    40    40

>> isnan(example)

ans =

     0     0     0     0
     0     0     0     0
     0     0     0     0
```

READ/WRITE DATA

Exporting Data to Excel

xlswrite *('filename',A):* writes A to first excel sheet of file "filename"

xlswrite *('filename',A,sheet):* writes to the specified sheet.

xlswrite *('filename',A,range):* writes to the specified range of first sheet.

xlswrite *('filename',A,sheet,range):* writes to the specified sheet and range

READ/WRITE DATA

```
>> A=[1 2 3;4 5 6;7 8 9];
>> xlswrite('example',A)
>> B=A*A;
>> xlswrite('example',B,2)
>> C=A(:,1);
>> xlswrite('example',C,3,'D1:D3')
```

READ/WRITE DATA

Importing Data from Excel

[num,txt,raw] = xlsread *(filename):* reads data from first excel sheet of file “filename”

[num,txt,raw] = xlsread *(filename,sheet):* reads data from specified sheet.

[num,txt,raw] = xlsread *(filename,range):* reads data from specified range of first sheet.

[num,txt,raw] = xlsread*(filename,sheet,range):* reads data from specified sheet an

READ/WRITE DATA

```
>> xlsread('example')

ans =

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

```
>> xlsread('example',2)

ans =

    30    36    42
    66    81    96
   102   126   150
```

```
>> xlsread('example',3,'D1:D3')

ans =

     1
     4
     7
```

READ/WRITE DATA

Xls vs. Csv

Xls

- holds data in worksheets, charts, and macros
- can be used in Windows

Csv (comma seperated values):

- set of file formats used to store tabular data in which numbers and text are stored
- more common in computer sciences
- both Linux & Windows can operate or

READ/WRITE DATA

Exporting /Importing Data (csv files)

csvwrite *('filename',A):* writes matrix A into filename as comma-separated values

csvread*('filename'):* reads a comma-separated value formatted file, filename

STATISTICAL ANALYSIS

Measures of Location

mean(x)

median(x):

mode(x)

Note that;

mean is the average of the data

median is the middle value of the data sorted by value

mode is the most common value

STATISTICAL ANALYSIS

```
>> a=[1 2 3;1 3 4;2 2 5]

a =

     1     2     3
     1     3     4
     2     2     5
```

```
>> mean(a)

ans =

    1.3333    2.3333    4.0000

>> median(a)

ans =

     1     2     4

>> mode(a)

ans =

     1     2     3
```

STATISTICAL ANALYSIS

Measures of Scale (Maximum and Minimum)

max *(x):* find the largest value in x

[y,k]=max*(x):* find y and k that are the maximum value of x, indices of first maximum.

max*(x,y)*: compares x and y and report the minimum

min*(x):* find the smallest value in x

[y,k]=min*(x):* find y and k that are the minimum value of x, indices of first minimum.

min*(x,y):* compares x and y and report the minimum

STATISTICAL ANALYSIS

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

```
>> max(a)
```

```
>> [y,k]=max(a)
```

```
>> min(a,b)
```

```
>> max(1,5)
```

STATISTICAL ANALYSIS

Measures of scale (Sum and Product)

sum *(x):* sum of elements

prod *(x):* product of elements

cumsum *(x):* cumulative sum of elements

cumprod *(x):* cumulative product of the elemensts

STATISTICAL ANALYSIS

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

```
>> sum(A)

ans =

    12    15    18
```

```
>> cumsum(A)

ans =

     1     2     3
     5     7     9
    12    15    18
```

```
>> prod(A)

ans =

    28    80   162
```

```
>> cumprod(A)

ans =

     1     2     3
     4    10
    28    80
```

STATISTICAL ANALYSIS

Measures of Scale (Standard deviation and Variance)

std *(x):* standard deviation of x

var *(x):* variance of x

Note that;

standard deviation shows how much variation there is from the data average

variance shows how far a set of numbers are spread out from each other.

DATA VISUALIZATION

Histogram

hist *(x):* histogram of x using 10 bins

by vector y

hist *(x,nbins):* histogram of x using nbins bins

hist *(x,y):* histogram of x with bin centers specified

DATA VISUALIZATION

```
>> A=[1 2 3];
>> [n,xc]=hist(A)

n =

     1     0     0     0     1     0     0     0     0     1


xc =

  Columns 1 through 7

    1.1000    1.3000    1.5000    1.7000    1.9000    2.1000    2.3000

  Columns 8 through 10

    2.5000    2.7000    2.9000

>>
```

Divides the data to 10 bins

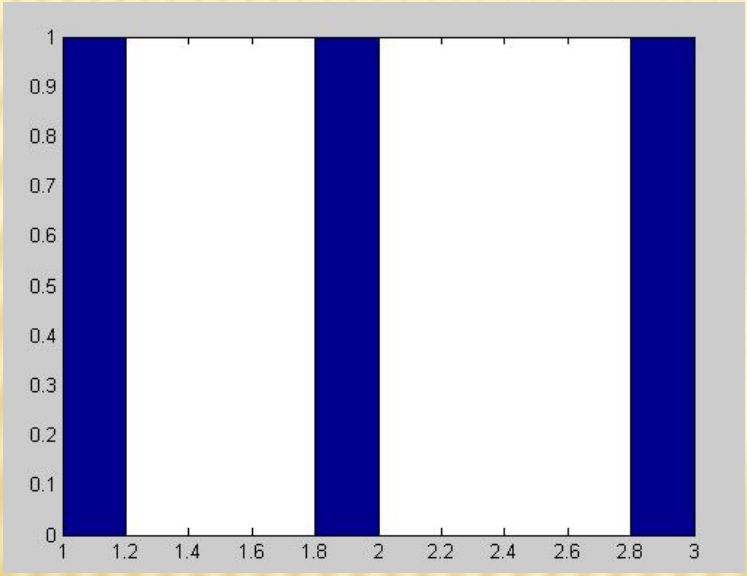
number of data corresponding to bins

bin centers

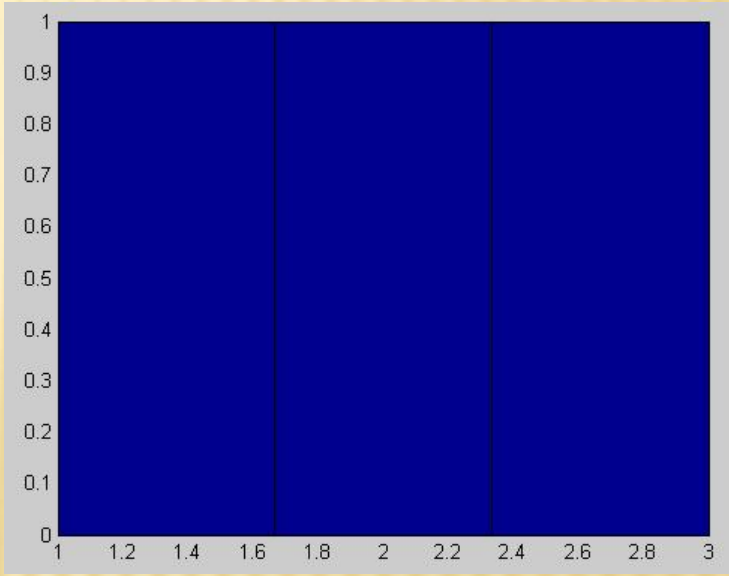
Bin1	0-1.2
Bin2	1.2-1.4
Bin3	1.4-1.6
Bin4	1.6-1.8
Bin5	1.8-2.0
Bin6	2.0-2.2
Bin7	2.2-2.4
Bin8	2.4-2.6
Bin9	2.6-2.8
Bin 10	2.8-3.0

DATA VISUALIZATION

```
>> hist(A)
```

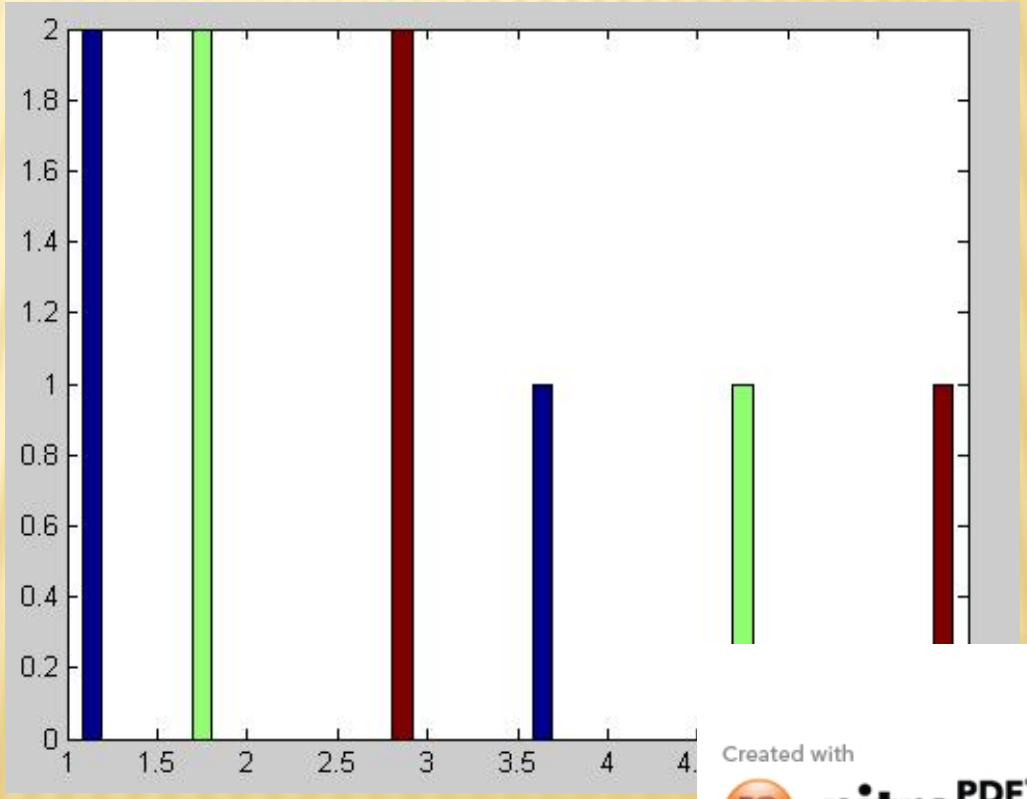



```
>> hist(A,3)
```



DATA VISUALIZATION

```
>> A=[1 2 3;1 2 3;4 5 6]

A =

     1     2     3
     1     2     3
     4     5     6
```



DATA VISUALIZATION

Bar Plot

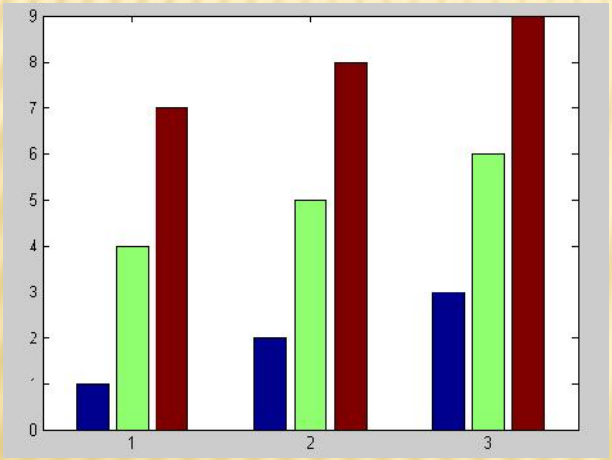
bar *(x):* draws one bar for each element of x

bar*(x,y):* draws one bar for each element of y where x is a vector defining x-axis intervals for vertical bar

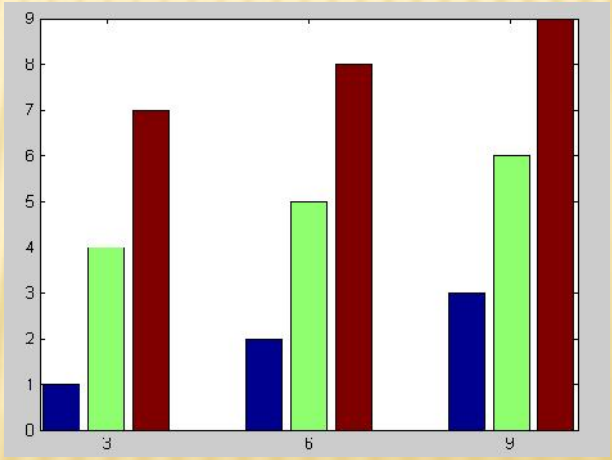
bar *(x,y,width):* width can be assigned (default=0.8)

DATA VISUALIZATION

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

a =

     1     4     7
     2     5     8
     3     6     9

>> bar(a)
```




```
>> b=[3 6 9];
>> bar(b,a)
```




```
>> bar(b,a,1.5)
```

DATA VISUALIZATION

2-D Scatter Plot

scatter *(x,y):* draws in default color and size

scatter *(x,y,S):* draws in default color with specified size.

scatter *(...,'filled'): fills the markers.*

3-D Scatter Plot

scatter3 *(x,y,z,...) :same as 2-D*

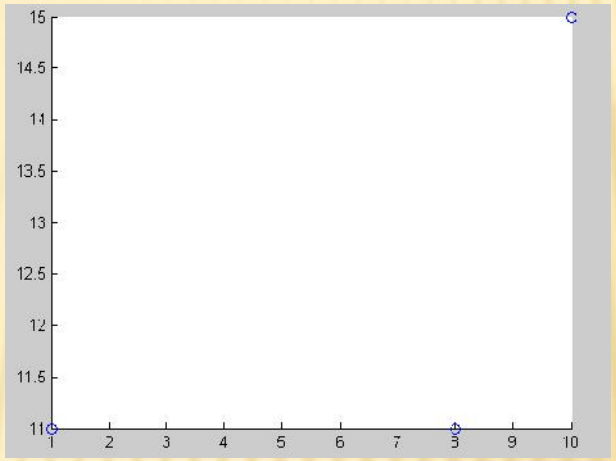
DATA VISUALIZATION

```
a =

     1    11     3
    10    15     4
     8    11     9
```

```
>> c=a(:,1)

c =

     1
    10
     8
```

```
>> d=a(:,2)

d =

    11
    15
    11
```

```
>> scatter (c,d)
```



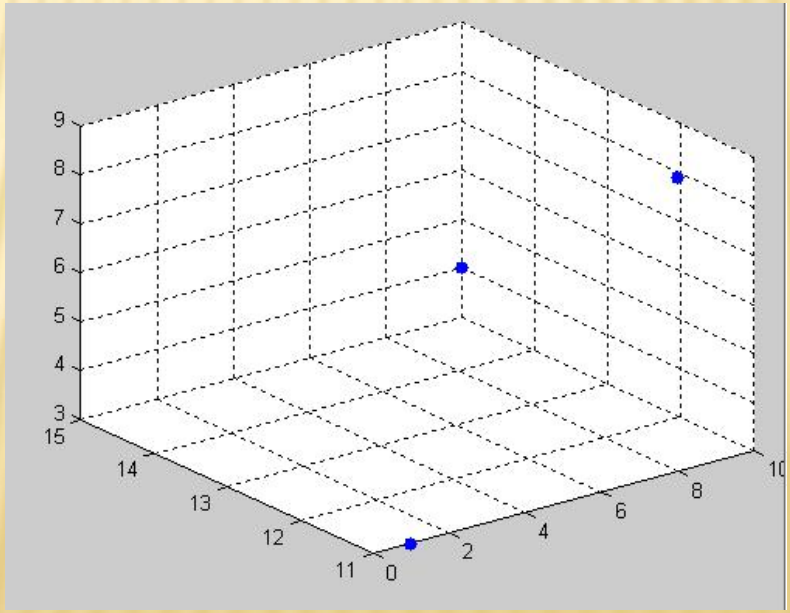

```
>> scatter (c,d,'filled')
```

DATA VISUALIZATION

```
>> c=a(:,1)

c =

     1
    10
     8
```

```
>> d=a(:,2)

d =

    11
    15
    11
```

```
>> e=a(:,3)

e =

     3
     4
     9
```



DATA VISUALIZATION

3-D Plot

meshgrid *(x,y):* transforms the domain specified by vectors x and y into arrays X and Y to evaluate functions of two variables and three-dimensional mesh/surface plots.

DATA VISUALIZATION

```
x =

     1     2     3

>> y=10:14

y =

    10    11    12    13    14
```

```
>> [X,Y]=meshgrid(x,y)

X =

     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3


Y =

    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
```

DATA VISUALIZATION

plot3 *(x,y,z): 3d plot of x,y and z.*

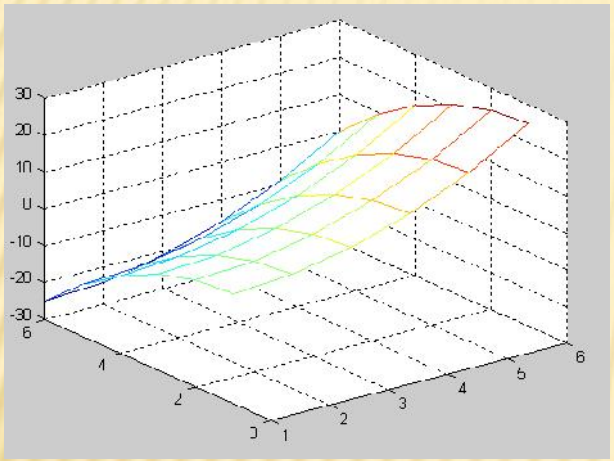
mesh *(x,y,z): wireframe mesh of x,y and z.*

surf *(x,y,z): shaded surface of x,y and z.*

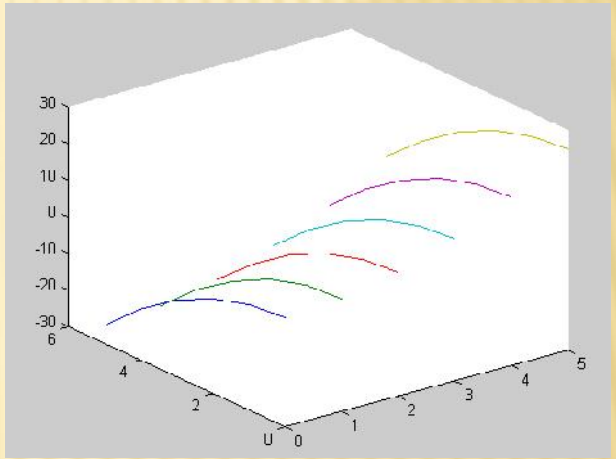
contour *(x,y,z): contour plot of z using x and y*

```
>> [x,y]=meshgrid(0:5);
>> z=x.^2-y.^2;
```

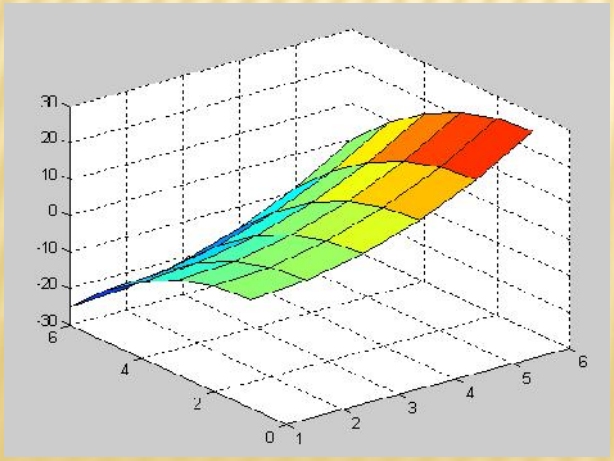
```
>> mesh(z)
```



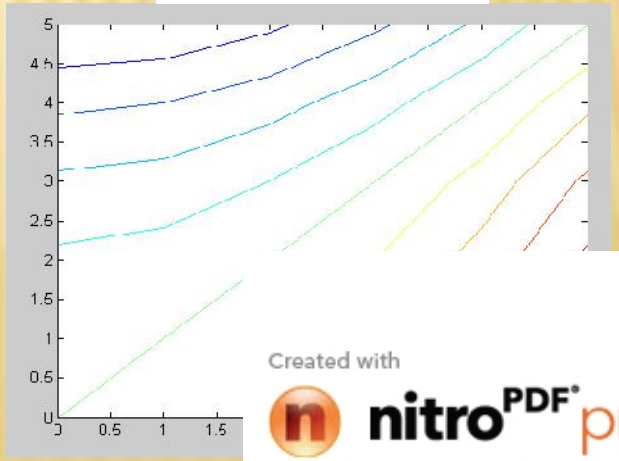
```
>> plot3(x,y,z)
```



```
>> surf(z)
```



```
>> contour(z)
```



SUMMARY

You should learn

How to read/write data

Working on data (statistical analysis)

Plotting Data (data visualization)

ANY QUESTIONS?