

# 12-Reading Data

text: Chapter 4.4-4.5

ECEGR 101

Engineering Problem Solving with Matlab

Professor Henry Louie



# Overview

- Reading Data from .txt
- Reading Data from .xls and .csv
- Reading Data from .mat
- User Interface Import



# Reading Text Data

To read text data from a text file

1. Open the file (**fopen**)
2. Read the variables from the file (**fscanf**)
3. Close the file (**fclose**)



# fopen

- **fid = fopen(FILENAME, PERMISSION)**

FILENAME: name of the file to open, including extension (e.g. .txt). File must be in current directory. You can also specify the path to the file.

PERMISSION (optional): opens FILENAME in the following mode

'r' open file for reading

'w' open file for writing; discard existing contents

'a' open or create file for writing; append data to end of file

See help fopen for more details



# fopen

- Example

```
>> fid_1 = fopen('reading_data_example1.txt','r')  
  
fid_1 =  
  
      7  
  
>> fid_2 = fopen('reading_data_example2.txt','r')  
  
fid_2 =  
  
      8
```

Opening two different files,  
and giving their file IDs  
different variable names,  
fid\_1, and fid\_2

```
>> whos  
  
Name          Size          Bytes  Class  Attributes  
  
fid_1         1x1           8  double  
fid_2         1x1           8  double
```

fid is just a number that  
identifies a file



# fscanf Function

`array = fscanf(fid, format)`

or

`array = fscanf(fid, format, size)`

fid: file identification

format: controls how the data is read, for example  
`'%d %f'`

Size: the amount of data to be read from the file

`n` – read exactly `n` values

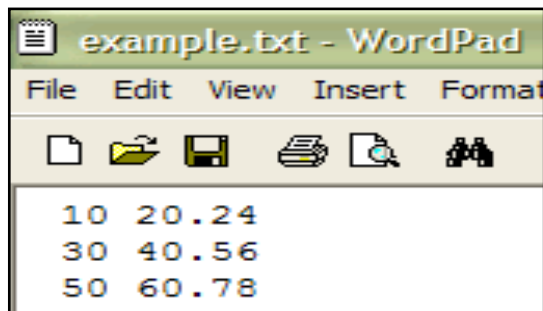
`inf` – read until the end of the file

`[n m]` – read exactly `n x m` values, and format the data as an `n x m` array.



# Example

- Read the following file: (saved as reading\_data\_example1.txt)
- Save data as "z"



```
fid_1 = fopen('reading_data_example1.txt','r');  
z = fscanf(fid_1,'%d %f')  
fclose(fid_1)
```

```
>> z  
z =  
    10.0000  
    20.2400  
    30.0000  
    40.5600  
    50.0000  
    60.7800  
>>
```

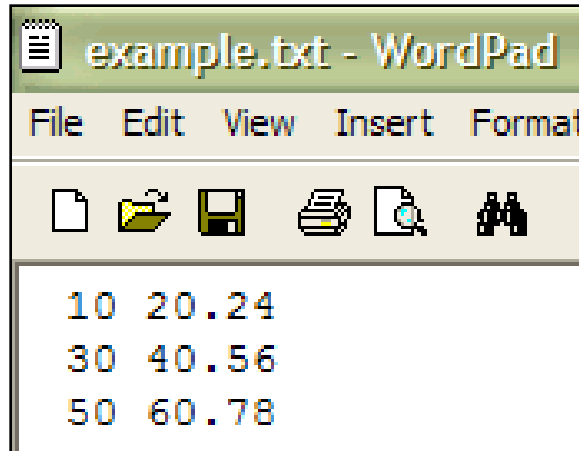
%f or %d    %f

How do we preserve rows and columns?



# Example

Try this



```
fid_1 = fopen('reading_data_example1.txt','r');  
z = fscanf(fid_1,'%d %f',[3 2])  
fclose(fid_1)
```

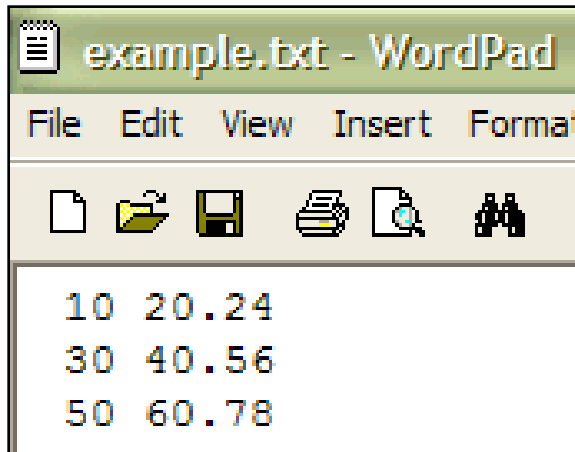
```
>> z  
z =  
    10.0000    40.5600  
    20.2400    50.0000  
    30.0000    60.7800  
>>
```

Not quite right.





# Example



```
fid_1 = fopen('reading_data example1.txt','r');
z = fscanf(fid_1,'%d %f' [2 3])
fclose(fid_1)
```

Text file is read row-wise, while the output matrix is filled column-wise.

```
>> z
z =
    10.0000    30.0000    50.0000
    20.2400    40.5600    60.7800
>>
>> z = z';
>> z
z =
    10.0000    20.2400
    30.0000    40.5600
    50.0000    60.7800
>>
```



# Exercise

Write a script to read the data in the file 'reading\_data\_example2.txt'. Store the data as a variable named 'lottery\_numbers'. Make sure the number of rows and columns are preserved.



# Exercise

```
reading_data_example2 - Notepad
File Edit Format View Help
6 15 19 23 40 5
4 16 24 25 44 5
9 23 27 49 51 38
4 11 32 39 40 33
1 15 20 21 47 34
6 15 27 31 39 25
9 22 28 48 54 8
```

Data is %d (or %f) and has 7 rows,  
and 6 columns

```
fid_2 = fopen('reading_data_example2.txt','r');
lottery_numbers = fscanf(fid_2, '%d %d', [6 7]);
fclose(fid_2)
```

use [6 7] and transpose

```
lottery_numbers =

     6     15     19     23     40     5
     4     16     24     25     44     5
     9     23     27     49     51    38
     4     11     32     39     40    33
     1     15     20     21     47    34
     6     15     27     31     39    25
     9     22     28     48     54     8
```

7 x 6



# Reading Data from .csv, .xls

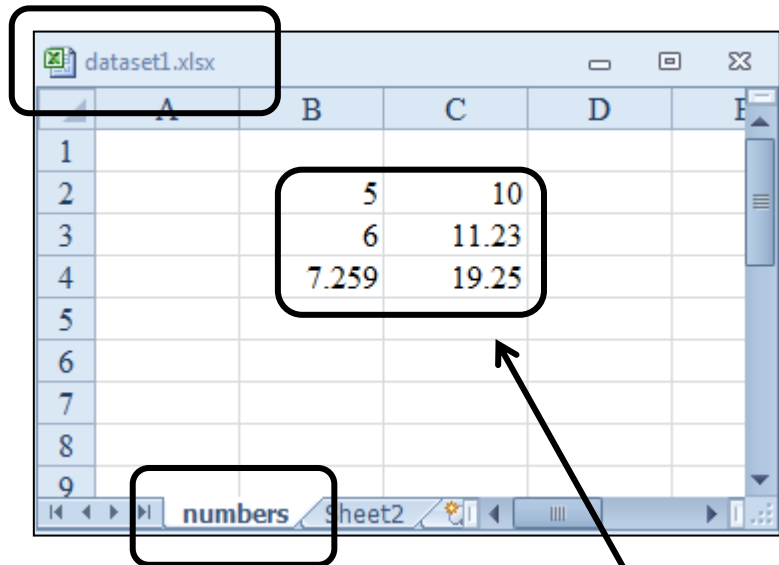
- Data can be read from .csv, .xls, .xlsx files
- `[data]=xlsread(FILE, SHEET, RANGE)`
  - data: variable containing numeric values
  - FILE: name of file, must include extension
  - SHEET: worksheet in the file to be read (optional)
  - RANGE: range of data in the worksheet to be read (optional)

File to be read must be in working directory



# Example

filename



```
>> ls
.          dataset1.xlsx
..         demolition.jpg
EV_data.xlsx price.mat

>> values=xlsread('dataset1.xlsx','numbers','B2:C4')
values =
    5.0000    10.0000
    6.0000    11.2300
    7.2590    19.2500
```

worksheet

Upper left cell: B2  
Lower right cell: C4



## Exercise

- The file "EV\_data.xlsx" contains data samples of the electricity consumed by electric vehicles (EVs) being charged in the Seattle area. The data corresponds to 15-minute averages.
- Read the data in the average kilowatt (aKW) column of the file 'EV\_data.xlsx' in sheet "Format 1" for 1/1/11 0:00 through 12/31/12 23:45. Store the data as the variable "demand".

Hint: open the file in Excel to see what the data range should be



## Exercise

- `demand=xlsread('EV_data.xlsx','Format 1','B5:B70180');`

It is a good idea to suppress the output if importing a large amount of data



## Exercise

- How many data points are contained in “demand”?
- What is the average amount of power consumed by the EVs?
- What are the maximum and minimum amount of power consumed by the EVs?
- How many kilowatthours of energy are consumed by the EVs?





## Exercise

- How many data points are contained in “demand”?
- What is the average amount of power consumed by the EVs?
- What are the maximum and minimum amount of power consumed by the EVs?
- How many kilowatthours of energy are consumed by the EVs?



## Exercise

- How many data points are contained in “demand”?

```
>> size(demand)
ans =
    35040     1
```

- What is the average amount of power consumed by the EVs?

```
>> mean(demand)
ans =
    59.6211
```

 59.62 kW

- What are the maximum and minimum amount of power consumed by the EVs?

```
>> max(demand)
ans =
    422.7685
>> min(demand)
ans =
     0
```

- How many kilowatthours of energy are consumed by the EVs?

```
>> format shorteng
>> sum(demand)/4
ans =
    522.2805e+003
```

 522.28 MWh



# xlsread

- See help xlsread for information on reading text data
- xlsread can be slow
  - minimize use
  - Read data once, and store as .mat (use 'save' command) for future use
- How does xlsread handle an empty cell?



# Reading Data from .mat

- Variables and their content can be stored and retrieved in the .mat format
- “load FILE” retrieves all variables saved in FILE
- Possible to specify particular variables in the FILE



# Example

## Variables to be loaded

```
>> clear all; load player
>> who

Your variables are:

HR          RBI          batting_avg  name

>> name
name =
Brett Boone
>> batting_avg
batting_avg =
    0.2660
```

All four variables are loaded into the workspace

```
>> clear all; load player batting_avg HR
>> who

Your variables are:

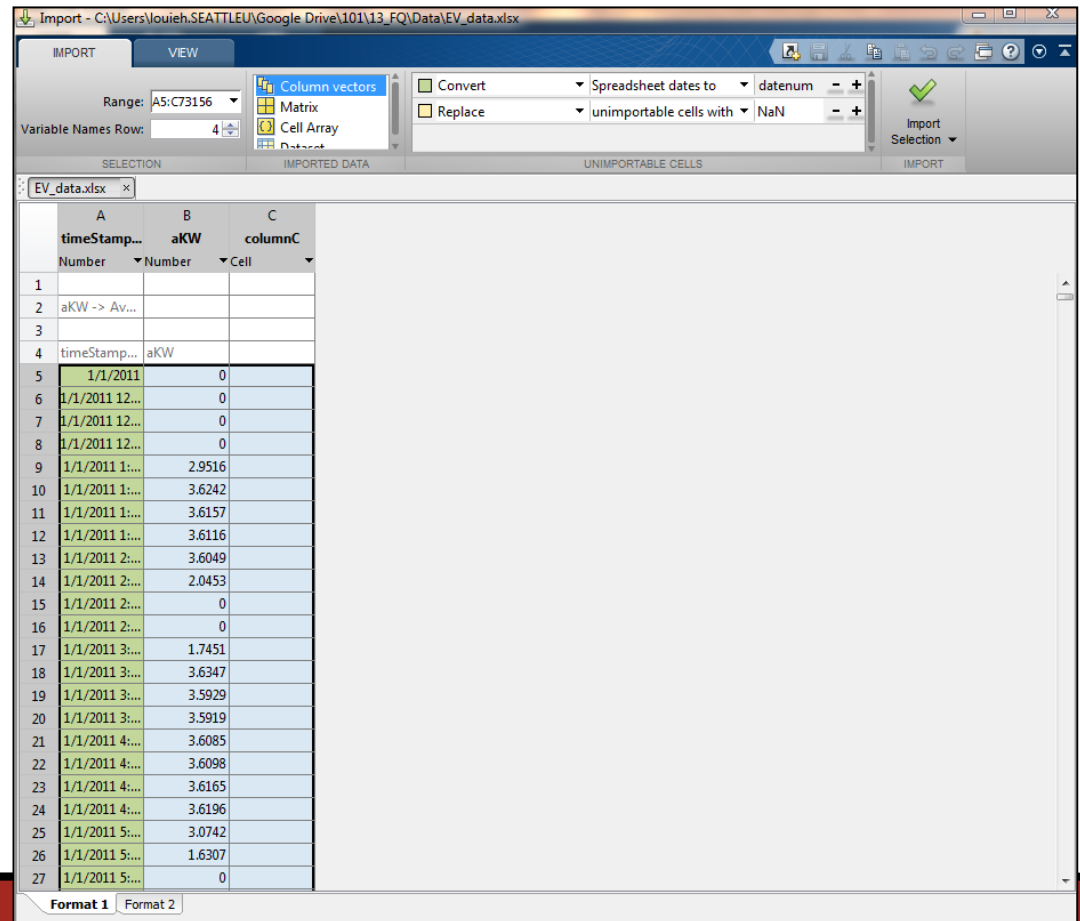
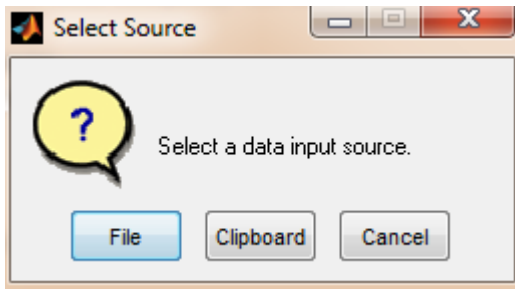
HR          batting_avg
```

Only specified variables are loaded into the workspace



# Importing Data

- GUI (graphical user interface) data importing tool
- Type "uiimport"





# Raw Files

- It also is possible to write and read **binary files** in MATLAB (**fread** and **fwrite**)