

# Computer Programming with MATLAB



## Lesson 8: File Input/Output

by

Akos Ledeczki and Mike Fitzpatrick

# File Input/Output

## ▶ File:

- Area in permanent storage (disk drive)
- Stores information
- Managed by the operating system
- Can be copied or moved
- Can be accessed by programs

## ▶ File Input/Output (I/O)

- Data exchange between programs and computers
- Data exchange between the physical world and computers
- Saving your work so you can continue with it later

## ▶ MATLAB can handle

- Mat-files and M-files AND text, binary, and Excel files

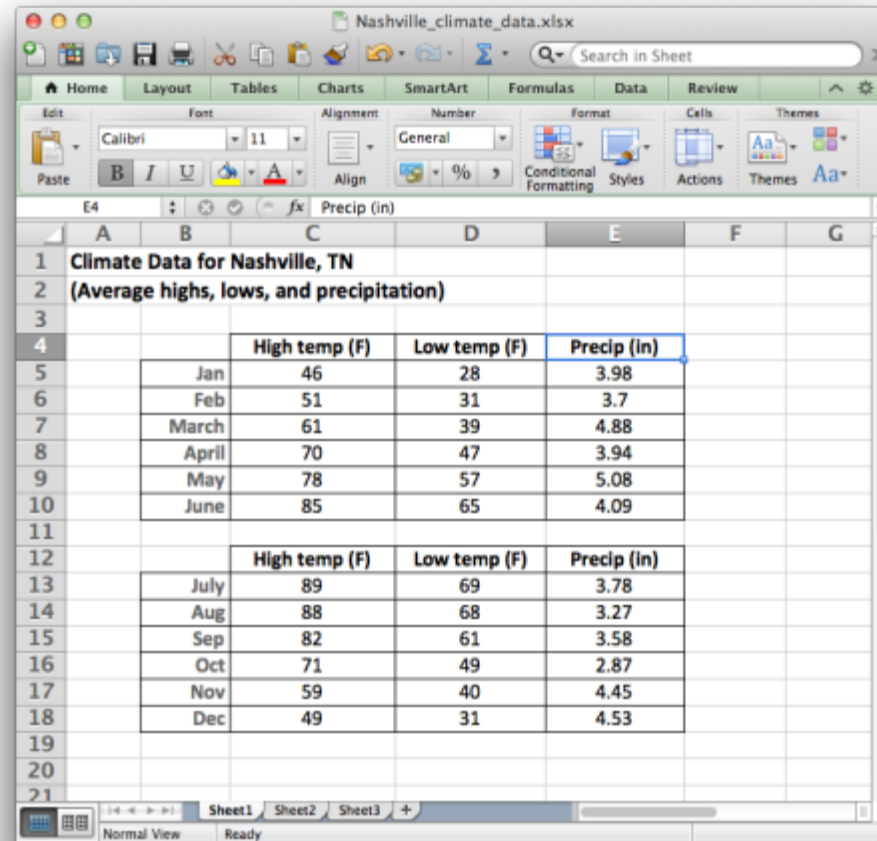


VANDERBILT  
UNIVERSITY

# Excel files

- ▶ Microsoft Excel® is a widely used data-analysis tool
- ▶ Many other programs support reading and writing Excel files
- ▶ MATLAB does too with two built-in functions
  - `xlsread`
  - `xlswrite`

# Reading Excel files



Climate Data for Nashville, TN (Average highs, lows, and precipitation)				
		High temp (F)	Low temp (F)	Precip (in)
Jan		46	28	3.98
Feb		51	31	3.7
March		61	39	4.88
April		70	47	3.94
May		78	57	5.08
June		85	65	4.09
		High temp (F)	Low temp (F)	Precip (in)
July		89	69	3.78
Aug		88	68	3.27
Sep		82	61	3.58
Oct		71	49	2.87
Nov		59	40	4.45
Dec		49	31	4.53

```
>> [num,txt,row] = xlsread('Nashville_climate.xlsx');
```

# Numerical



VANDERBILT  
UNIVERSITY

```
>> num
```

```
num =
```

46	28	3.98
51	31	3.7
61	39	4.88
70	47	3.94
78	57	5.08
85	65	4.09
NaN	NaN	NaN
NaN	NaN	NaN
89	69	3.78
88	68	3.27
82	61	3.58
71	49	2.87
59	40	4.45
49	31	4.53

The screenshot shows an Excel spreadsheet titled "Nashville\_climate\_data.xlsx". The spreadsheet contains climate data for Nashville, TN, including average high and low temperatures and precipitation for each month. The data is organized into two tables, one for the first half of the year (January to June) and one for the second half (July to December). The first table has columns for High temp (F), Low temp (F), and Precip (in). The second table also has columns for High temp (F), Low temp (F), and Precip (in). The data is as follows:

	High temp (F)	Low temp (F)	Precip (in)
Jan	46	28	3.98
Feb	51	31	3.7
March	61	39	4.88
April	70	47	3.94
May	78	57	5.08
June	85	65	4.09
July	89	69	3.78
Aug	88	68	3.27
Sep	82	61	3.58
Oct	71	49	2.87
Nov	59	40	4.45
Dec	49	31	4.53

# Numerical



VANDERBILT  
UNIVERSITY

```
>> num
```

```
num =
```

46	28	3.98
51	31	3.7
61	39	4.88
70	47	3.94
78	57	5.08
85	65	4.09
NaN	NaN	NaN
NaN	NaN	NaN
89	69	3.78
88	68	3.27
82	61	3.58
71	49	2.87
59	40	4.45
49	31	4.53

Nashville\_climate\_data.xlsx

Search in Sheet

Home Layout Tables Charts SmartArt Formulas Data Review

fx Precip (in)

	A	B	C	D	E	F	G
1	Climate Data for Nashville, TN						
2	(Average highs, lows, and precipitation)						
3							
4			High temp (F)	Low temp (F)	Precip (in)		
5		Jan	46	28	3.98		
6		Feb	51	31	3.7		
7		Mar	61	39	4.88		
8		Apr	70	47	3.94		
9		May	78	57	5.08		
10		Jun	85	65	4.09		
11							
12			High temp (F)	Low temp (F)	Precip (in)		
13		Jul	89	69	3.78		
14		Aug	88	68	3.27		
15		Sep	82	61	3.58		
16		Oct	71	49	2.87		
17		Nov	59	40	4.45		
18		Dec	49	31	4.53		
19							
20							
21							

Sheet1 Sheet2 Sheet3

Normal View Ready

# Numerical



VANDERBILT  
UNIVERSITY

```
>> num
```

```
num =
```

46	28	3.98
51	31	3.7
61	39	4.88
70	47	3.94
78	57	5.08
85	65	4.09
NaN	NaN	NaN
NaN	NaN	NaN
89	69	3.78
88	68	3.27
82	61	3.58
71	49	2.87
59	40	4.45
49	31	4.53

Nashville\_climate\_data.xlsx

Search in Sheet

Home Layout Tables Charts SmartArt Formulas Data Review

Edit Font Alignment Number Format Cells Themes

Paste B I U A Align General Conditional Formatting Styles Actions Themes Aa

E4 fx Precip (in)

	A	B	C	D	E	F	G
1	Climate Data for Nashville, TN						
2	(Average highs, lows, and precipitation)						
3							
4			High temp (F)	Low temp (F)	Precip (in)		
5	Jan	46	28	3.98			
6	Feb	51	31	3.7			
7	Mar	61	39	4.88			
8	Apr	70	47	3.94			
9	May	78	57	5.08			
10	Jun	85	65	4.09			
11	Jul	89	69	3.78			
12	Aug	88	68	3.27			
13	Sep	82	61	3.58			
14	Oct	71	49	2.87			
15	Nov	59	40	4.45			
16	Dec	49	31	4.53			
17							
18							
19							
20							
21							

Sheet1 Sheet2 Sheet3

Normal View Ready

# Text



VANDERBILT  
UNIVERSITY

```
>> txt =
```

```
txt =
```

```
[1x30 char] ''      ''      ''      ''
[1x40 char] ''      ''      ''      ''
''          ''      ''      ''      ''
''          ''      'High temp (F)' 'Low temp (F)' 'Precip (in)'
```

''	'Jan'	''	''	''
''	'Feb'	''	''	''
''	'March'	''	''	''
''	'April'	''	''	''
''	'May'	''	''	''
''	'June'	''	''	''
''	''	''	''	''
''	''	'High temp (F)'	'Low temp (F)'	'Precip (in)'
''	'July'	''	''	''
''	'Aug'	''	''	''
''	'Sep'	''	''	''
''	'Oct'	''	''	''
''	'Nov'	''	''	''
''	'Dec'	''	''	''



# All data: cell array



VANDERBILT  
UNIVERSITY

```
>> raw
```

```
raw =
```

```
[1x30 char] [ NaN] [ NaN]          [ NaN]          [ NaN]
[1x40 char] [ NaN] [ NaN]          [ NaN]          [ NaN]
[ NaN]      [ NaN] [ NaN]          [ NaN]          [ NaN]
[ NaN]      [ NaN] 'High temp (F)' 'Low temp (F)' 'Precip (in)'
```

[ NaN]	'Jan'	[ 46]	[ 28]	[ 3.98]
[ NaN]	'Feb'	[ 51]	[ 31]	[ 3.7]
[ NaN]	'March'	[ 61]	[ 39]	[ 4.88]
[ NaN]	'April'	[ 70]	[ 47]	[ 3.94]
[ NaN]	'May'	[ 78]	[ 57]	[ 5.08]
[ NaN]	'June'	[ 85]	[ 65]	[ 4.09]
[ NaN]	[ NaN]	[ NaN]	[ NaN]	[ NaN]
[ NaN]	[ NaN]	'High temp (F)'	'Low temp (F)'	'Precip (in)'
[ NaN]	'July'	[ 89]	[ 69]	[ 3.78]
[ NaN]	'Aug'	[ 88]	[ 68]	[ 3.27]
[ NaN]	'Sep'	[ 82]	[ 61]	[ 3.58]
[ NaN]	'Oct'	[ 71]	[ 49]	[ 2.87]
[ NaN]	'Nov'	[ 59]	[ 40]	[ 4.45]
[ NaN]	'Dec'	[ 49]	[ 31]	[ 4.53]



VANDERBILT  
UNIVERSITY

# Text files

- ▶ Text files contain characters
- ▶ They use an encoding scheme:
  - ASCII or
  - Any one of many other schemes
  - MATLAB takes care of encoding and decoding
- ▶ Before using a text file, we need to open it
- ▶ Once done with the file, we need to close it

# Opening text files

- ▶ Opening: `fid = fopen(filename, permission)`
- ▶ Closing: `fclose(fid)`
- ▶ `fid`: Unique file identifier for accessing file
- ▶ Permission: what we want to do with the file—
  - read, write, overwrite, append, etc.

2ND ARGUMENT	PERMISSION
'rt'	open text file for reading
'wt'	open text file for writing; discard existing contents
'at'	open or create text file for writing; append data to end of file
'r+t'	open (do not create) text file for reading and writing
'w+t'	open or create text file for reading and writing; discard existing contents
'a+t'	open or create text file for reading and writing; append data to end of file



VANDERBILT  
UNIVERSITY

# Reading text files

- ▶ One line at a time
- ▶ `type` prints a text file in the command window
- ▶ Let's re-implement it:

```
function view_text_file(filename)
fid = fopen(filename,'rt');
if fid < 0
    error('error opening file %s\n\n', filename);
end

% Read file as a set of strings, one string per line:
oneline = fgets(fid);
while ischar(oneline)
    fprintf('%s',oneline) % display one line
    oneline = fgets(fid);
end
fprintf('\n');
fclose(fid);
```

# Reading text files

- ▶ Reading lines into string variables is easy
- ▶ Parsing these strings to get numerical data is much harder
- ▶ Not covered
- ▶ Binary files are more suited for numerical data

# Binary files

- ▶ Binary file = “not a text file”
- ▶ Many different ways to represent numbers
- ▶ All we need to know are their types.
- ▶ Binary files need to be
  - Opened with **fopen**
  - Closed with **fclose**

2ND ARGUMENT	PERMISSION
'r'	open binary file for reading
'w'	open binary file for writing; discard existing contents
'a'	open or create binary file for writing; append data to end of file
'r+'	open (do not create) binary file for reading and writing
'w+'	open or create binary file for reading and writing; discard existing contents
'a+'	open or create binary file for reading and writing; append data to end of file

# Writing binary files

- ▶ Data type is important
- ▶ Example: write a double array into a binary file

```
function write_array_bin(A,filename)
fid = fopen(filename,'w+');
if fid < 0
    error('error opening file %s\n', filename);
end

fwrite(fid,A,'double');

fclose(fid);
```

# Reading binary files

- ▶ Example: read a double array from a binary file

```
function A = read_bin_file(filename,data_type)
fid = fopen(filename,'r');
if fid < 0
    error('error opening file %s\n',filename);
end

A = fread(fid,inf,data_type);

fclose(fid);
```