File Input and Output

Chul Min Yeum and Jason Connelly

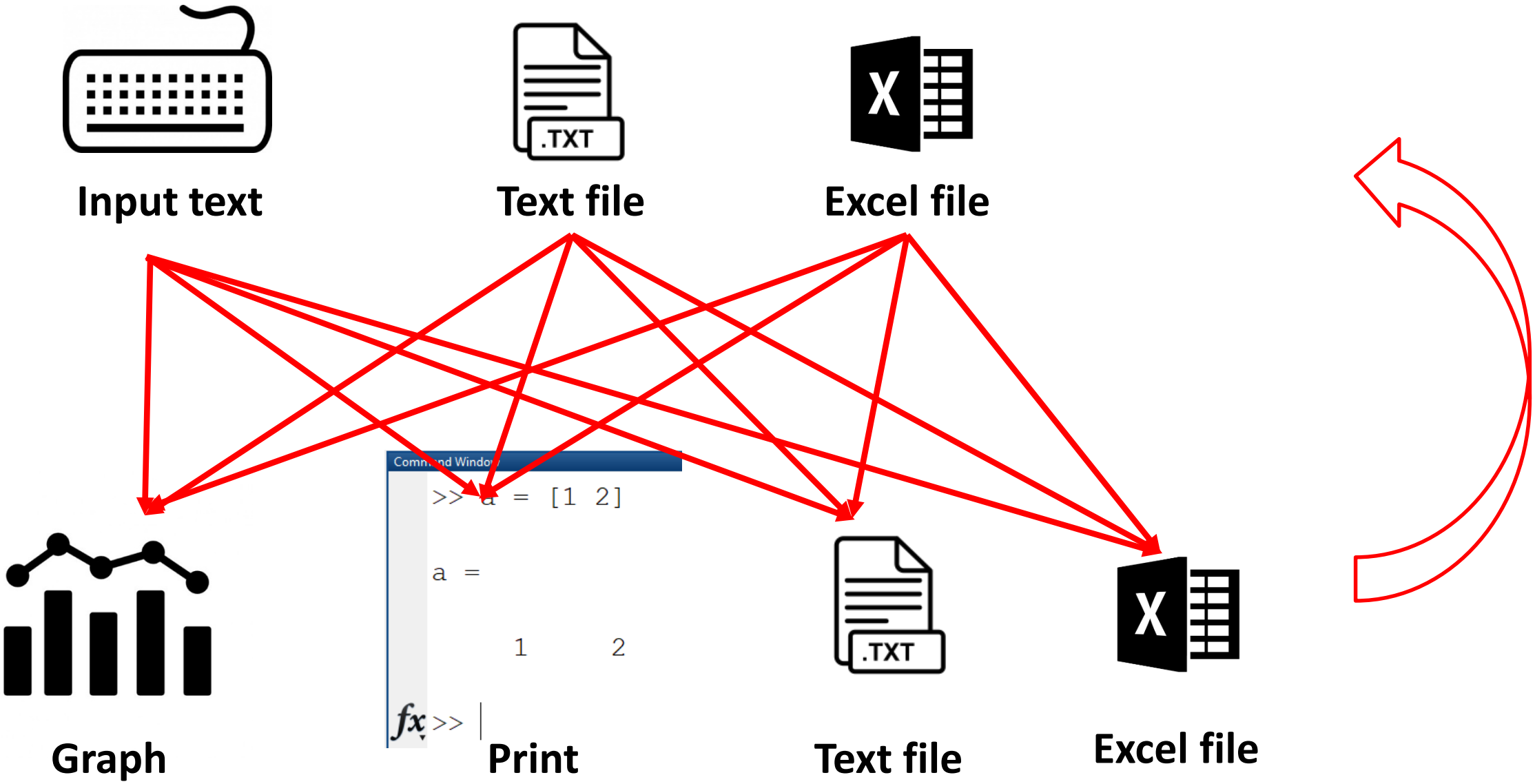
Civil and Environmental Engineering
University of Waterloo, Canada

AE121: Computational Method

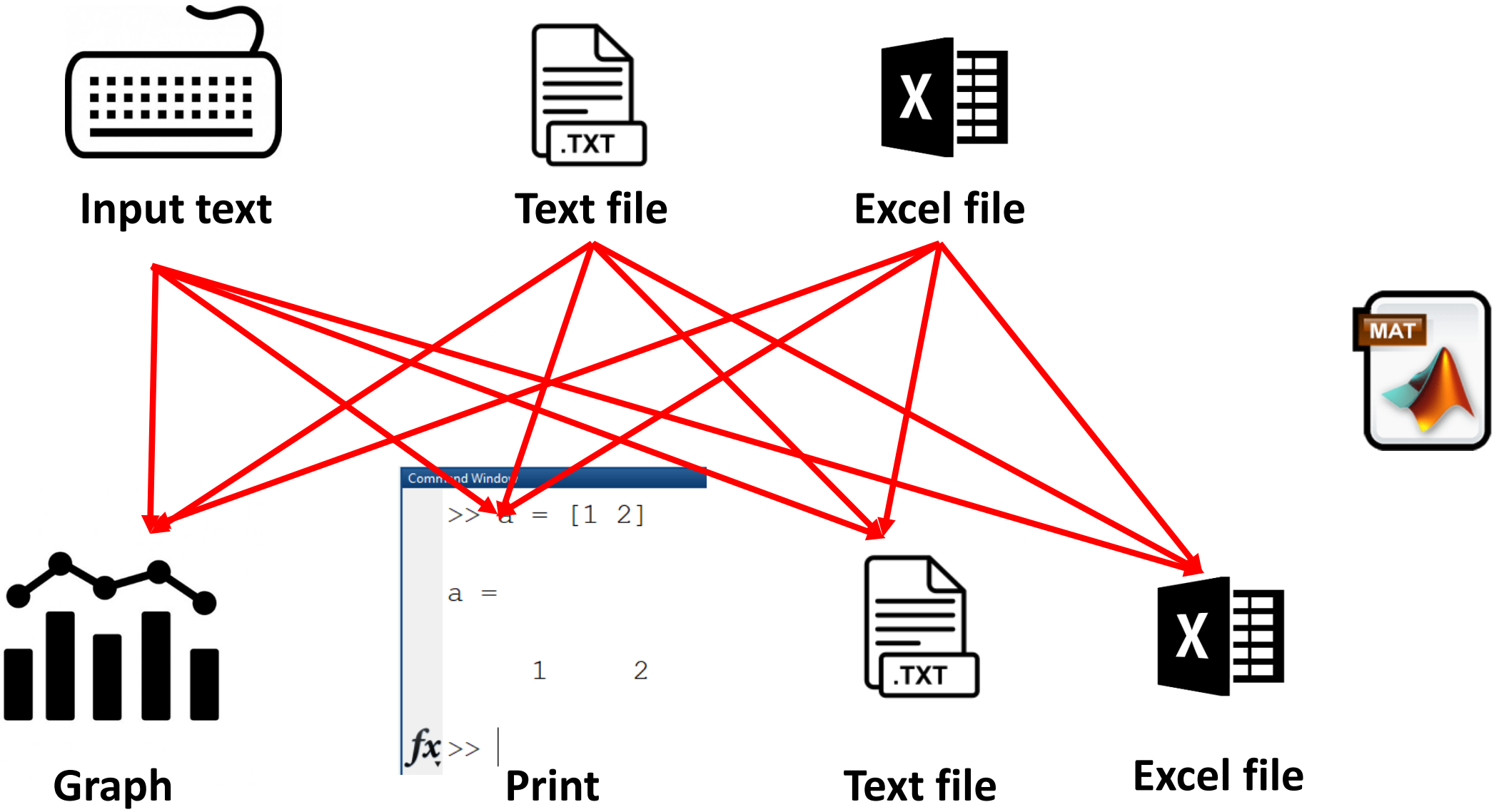
UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

Last updated: 2019-07-08

When Do We Use File Input and Output?



When Do We Use File Input and Output?



File Input & Output (I/O)

- Lower-level file I/O functions are used to read from, write to, and append to files when load and save cannot be used
- MATLAB has functions to read from and write to many different file types, for example, spreadsheets
- MATLAB has a special binary file type that can be used to store variables and their contents in MAT-files

Using MAT-files for Variables

- MATLAB has functions that allow reading and saving variables from files
- These files are called MAT-files (because the extension on the file name is .mat)
- Variables can be written to MAT-files, appended to them, and read from them
- Rather than just storing data, MAT-files store variable names and their values
- To save all workspace variables in a file, the command is:

  save filename
- To save just one variable to a file, the format is:

  save filename variablename
- To append:

  save -append filename variablename
- To read variables from a MAT-file into the base workspace:

  load filename variable list

Live Demo: File Save

```
clear; clc;
a = zeros(3,3);
b = randi(10,3,4);
c = randi(100,3,5);
d = randi(1000,1,30);

% save data_all
% load data_all a

% save data_all b
% load data_all

% save data_all
%
% clear; clc;
% load data_all a b
```

Importing Data from a .txt File

- The text file must be saved in the current folder that you are working in on MATLAB

- Delimiter: sequence of one or more characters used to specify boundaries between separate regions

- Three importing scenarios:
  1. Importing numeric data
  2. Importing character (string) data
  3. Importing numeric and character data

Print Formatting Operators

- The print formatting operators specify in what layout and what data type a column of data will be imported/exported as – for multiple columns of data, use multiple print format operators

- **Common Print Formatting Operators:**

  - %d –For integer numbers
  - %f –For floating point (decimal) numbers
  - %s –For entire character vectors or strings

- Note that you can only use **1 operator per column** (you cannot specify '%s' for the first value in a column and '%f' for all other values

- **Example:** If you wanted your first column of data to be integer data, the second column to be stored as character data and the third to be floating point numbers, the correct format string would be:

- '%d %s %f'

Revisit: Print Formatting

```
fprintf('Hello! Chul Min');
fprintf('Hello! \tChul Min');
fprintf('Hello! \nChul Min');
fprintf('Hello! AE121');
fprintf('Hello! %cE121', 'A');
fprintf('Hello! %s','AE121'); % string of characters
fprintf('Today is 04\\23\\2019.'); % two one slash to prin
fprintf('What is ''fprintf''?'); % two single quotes to pr
% Please copy above lines and paste them in the command wi
%
% comment: all outputs are placed on the same line (except the thrid one).
% This is because the live editor print the output based on the line of the
% code but the comand window (and editor) run them without adding new line.
% Please add \n at the end of your text to print ensuing output text on
% next line.
```

```
Hello! Chul Min
Hello!  Chul Min
Hello!
Chul Min
Hello! AE121
Hello! AE121
Hello! AE121
Today is 04\23\2019.
What is 'fprintf'?
```

textscan function

- The **textscan** function will create a cell array from the data read on the .txt file
- This function can be used to import numeric data, text data and both types of data from the same file easily
- When importing using textscan, the data is imported **column-by-column**
- You can specify using **print formatting operators** what type of data you want each column to be
- Data will be stored in a **cell,** where each column is a different cell array element
- You have the option of specifying a delimiter. The default delimiter is white-space.

fopen function

- Used to **open a file** for a specific purpose
- You can specify your purpose using a **permission specifier**, which is the second input
- The file does not have to already exist (although for reading data it should)
- The default permission is read only

Read only access:

```
fid = fopen('samp_data.txt', 'r');
fid = fopen('samp_data.txt');
```

Write only access (discard original contents of a file):

```
fid = fopen('new_data.txt', 'w');
```

Read and write access (discard original contents of file if writing):

```
fid = fopen('new_data.txt', 'w+');
```

General Procedure for Importing Data using textscan

1. **Open the file** to be read by creating a 'fid'
   - This is creating a hidden ID that represents the file

```
fid = fopen('samp_data.txt');
```

2. Use the **textscan function to import the data** using the proper format string and delimiter. If the delimiter you want is whitespace, you do not need to specify a delimiter

Comma delimiter:

```
samp_data = textscan(fid, '%f %f %f', 'delimiter', ',');
```

White-space delimiter:

```
samp_data = textscan(fid, '%f %f %f');
```

3. **Close the file** you have just read

```
fclose(fid);
```

Example: Columns of Numeric Data

samp_data.txt - Notepad

File Edit Format View Help

```
47, 0.95, 5
85, 1.12, 8
67, 0.83, 12
```

3 columns of data – when importing, you will need to specify **3 print formatting operators** if you want the grouping to remain the same

```
fid = fopen('samp_data.txt');

samp_data = textscan(fid, '%f %f %f', 'delimiter', ',')

col_2 = samp_data{2}

fclose(fid);
```

Command Window

```
samp_data =

  1×3 cell array

    {3×1 double}    {3×1 double}    {3×1 double}


col_2 =

    0.9500
    1.1200
    0.8300

fx >>
```

Question: Importing Text Data

countries_capitals.txt - Notepad

File Edit Format View Help

```
Canada, Ottawa
Kazakhstan, Astana
Mongolia, Ulaanbaatar
Papa New Guinea, Port Moresby
Rwanda, Kigali
South Korea, Seoul
Trinidad and Tobago, Port of Spain
```

Why isn't the white-space delimiter used in this case???

Example: Importing Text Data

```
fid = fopen('countries_capitals.txt');
geography_info = textscan(fid, '%s %s', 'delimiter', ',')
fclose(fid);

capitals = geography_info{2}
```

countries_capitals.txt - Notepad

File Edit Format View Help

```
Canada, Ottawa
Kazakhstan, Astana
Mongolia, Ulaanbaatar
Papa New Guinea, Port Moresby
Rwanda, Kigali
South Korea, Seoul
Trinidad and Tobago, Port of Spain
```

Command Window

```
geography_info =

  1×2 cell array

    {7×1 cell}    {7×1 cell}


capitals =

  7×1 cell array

    {'Ottawa'       }
    {'Astana'       }
    {'Ulaanbaatar'  }
    {'Port Moresby' }
    {'Kigali'       }
    {'Seoul'        }
    {'Port of Spain'}

fx >>
```

Example: Text and Numeric Data

N:\S2019\AE121\File IO\citydata.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

classes_W2020.txt | citydata.txt

```
Mexico City, 8854600
New York City, 8550405
Los Angeles, 3971883
Toronto, 2826498
Chicago, 2720546
Houston, 2296224
Havana, 2117625
Ecatepec de Morelos, 1760705
Montreal, 1753034
Philadelphia, 1567442
```

Normal text file | length : 209 lines : 10 | Ln : 1 Col : 1 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Example: Text and Numeric Data

```
fid = fopen('citydata.txt', 'r');
na_populations = textscan(fid, '%s %f', 'delimiter', ',');
fclose(fid);

cities = na_populations{1};

pop_data = na_populations{2};
```

```
na_populations =

  1×2 cell array

    {10×1 cell}    {10×1 double}
```

classes_W2020.txt | citydata.txt

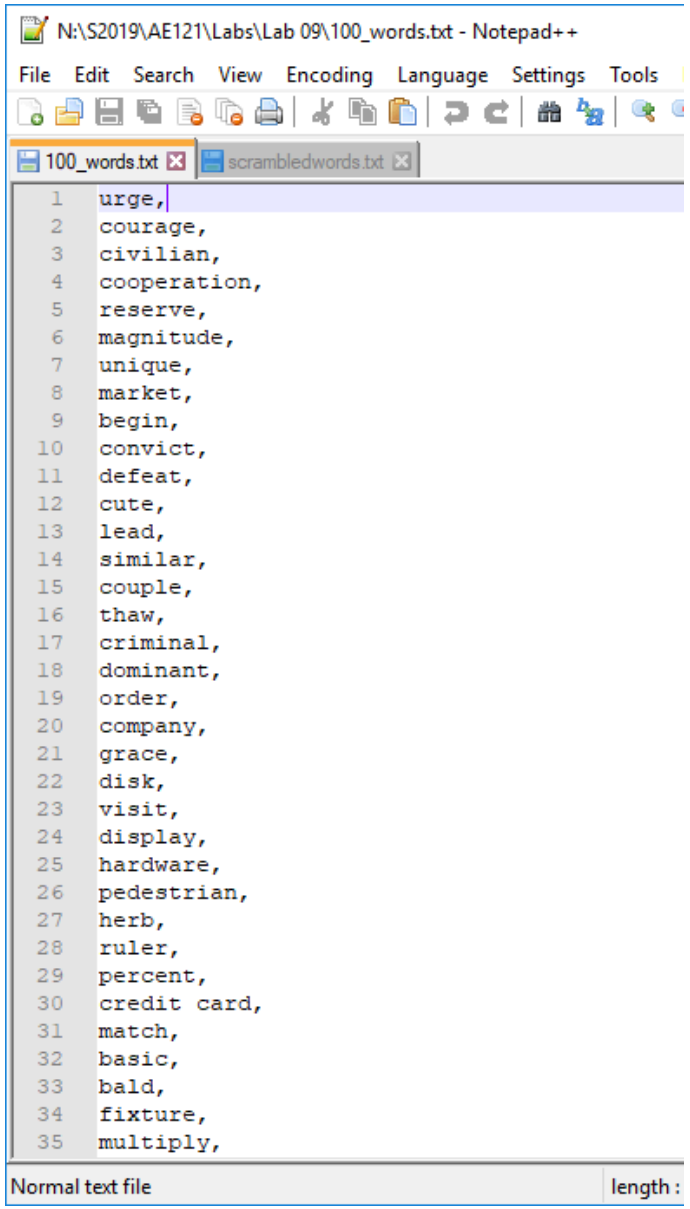
```
Mexico City, 8854600
New York City, 8550405
Los Angeles, 3971883
Toronto, 2826498
Chicago, 2720546
Houston, 2296224
Havana, 2117625
Ecatepec de Morelos, 1760705
Montreal, 1753034
Philadelphia, 1567442
```

Normal text file | length : 209 lines : 10 | Ln : 1 Col : 1 Sel : 0 | 0

```
cities =

  10×1 cell array

    {'Mexico City'        }
    {'New York City'      }
    {'Los Angeles'        }
    {'Toronto'            }
    {'Chicago'            }
    {'Houston'            }
    {'Havana'             }
    {'Ecatepec de Morelos'}
    {'Montreal'           }
    {'Philadelphia'       }
```

```
pop_data =

     8854600
     8550405
     3971883
     2826498
     2720546
     2296224
     2117625
     1760705
     1753034
     1567442
```

Example from Lab 09: Import Data from a .txt File



How to import a text file of words that have a comma as a delimiter?

```
fid = fopen('100_words.txt');
data_norm = textscan(fid, '%s', 'delimiter', ',')
fclose(fid);
```

How many columns are in this text file? What data type and size is 'data_norm'?

Example from Lab 09

N:\S2019\AE121\Labs\Lab 09\100_words.txt - Notepad++

File Edit Search View Encoding Language Settings Tools

100_words.txt | scrambledwords.txt

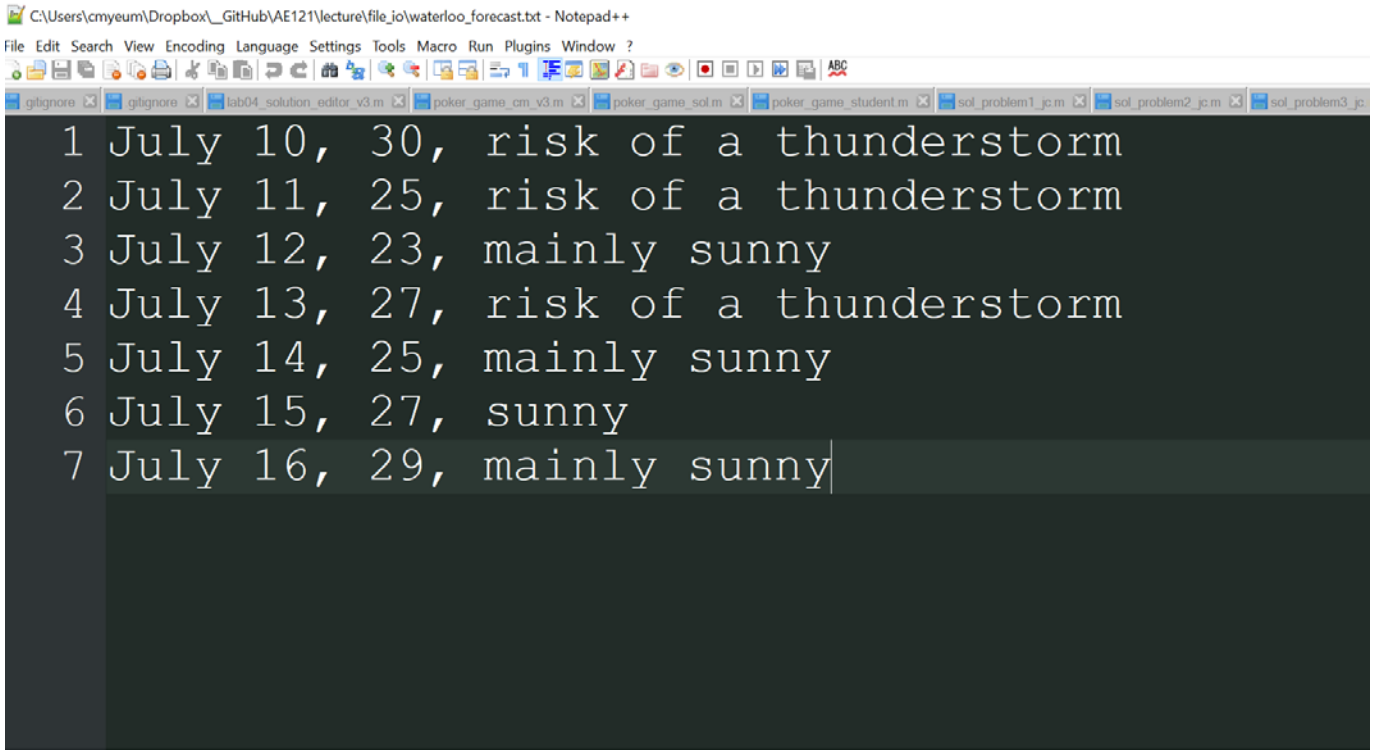
```
urge,
courage,
civilian,
cooperation,
reserve,
magnitude,
unique,
market,
begin,
convict,
defeat,
cute,
lead,
similar,
couple,
thaw,
criminal,
dominant,
order,
company,
grace,
disk,
visit,
display,
hardware,
pedestrian,
herb,
ruler,
percent,
credit card,
match,
basic,
bald,
fixture,
multiply,
```

Normal text file | length :

Command Window

```
>> data_norm

data_norm =

  1×1 cell array

    {100×1 cell}

>> data_norm{1}{25}

ans =

    'hardware'

>> data_norm{1}(25)

ans =

  1×1 cell array

    {'hardware'}
```

Demo


C:\Users\cmyeum\Dropbox__GitHub\AE121\lecture\file_io\waterloo_forecast.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
July 10, 30, risk of a thunderstorm
July 11, 25, risk of a thunderstorm
July 12, 23, mainly sunny
July 13, 27, risk of a thunderstorm
July 14, 25, mainly sunny
July 15, 27, sunny
July 16, 29, mainly sunny

Exporting Data to a .txt File

The fprintf function provides a way of exporting numeric or character data to a .txt file

```
fid = fopen('data_file.txt', 'w');
fprintf(fid, '%f %d\n', data);
fclose(fid);
```

General Procedure:

1 **Create a 'fid' :** The second input of fopen specifies what you want to do to the file. 'w' is write and discard existing contents, 'a' is write and append the text to the end of the file if there is already content in the file
https://www.mathworks.com/help/matlab/ref/fopen.html#btrnibn-1-permission

2 **Print data to the file using fprintf:** Print data using the correct **print format string** for the data you are writing (remember %d is for integers, %s for multiples characters or strings and %f is for floating-point numbers)
https://www.mathworks.com/help/matlab/ref/fprintf.html#btf8xsy-1_sep_shared-formatSpec

3 **Close the 'fid'**

fprintf Function

If you want data printed on different lines, you must include '\n' to move to the next line!

The file you create will be saved in the same folder you are currently working in

When using the fprintf function, input

fprintf(filename, formatSpec, data)

The data you want to print. This can be a scalar, vector or matrix of strings, doubles, characters etc.

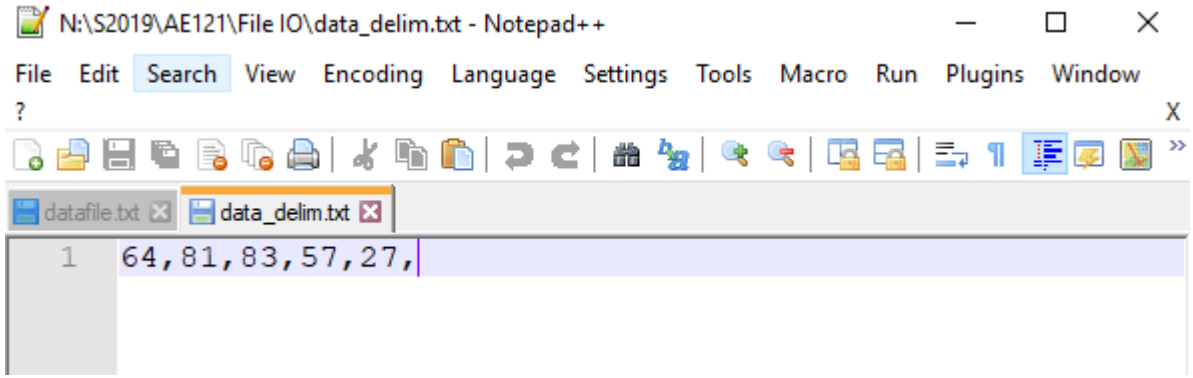
The filename you want to create

Your format string ex. '%d %f\n'

How would you add a delimiter to the text file you are writing?

How would You Add a Delimiter to the Text File You are Writing?

```
vec_data = randi(100,1,5)

fid = fopen('data_delim.txt', 'w');
fprintf(fid, '%d,', vec_data);% Row vector to text file
fclose(fid);
```



Example: Writing a Column Vector and Row Vector to a File

```
vec_data = randi(100,1,5)
fid = fopen('vecdata_file.txt', 'w');

fprintf(fid, '%d ', vec_data); % Row vector of 'vec_data' printed in text file
fprintf(fid, '\n\n', vec_data); % Make column vector start 2 lines later
fprintf(fid, '%d\n', vec_data); % Column vector of 'vec_data' printed in text file
fclose(fid);
```

Command Window

```
vec_data =

    86    53    83    11    62

>>
```

datafile.txt

```
86 53 83 11 62

86
53
83
11
62
```

Ln : 1 Col : 1 Sel : 0 | 0 Unix (LF) UTF-8 INS

fprintf with Matrices

- When you input a matrix to print, **<u>the order of the items will be printed follows linear indexing</u>**

- That means all elements in the first column of a matrix will be printed before any element in the second column

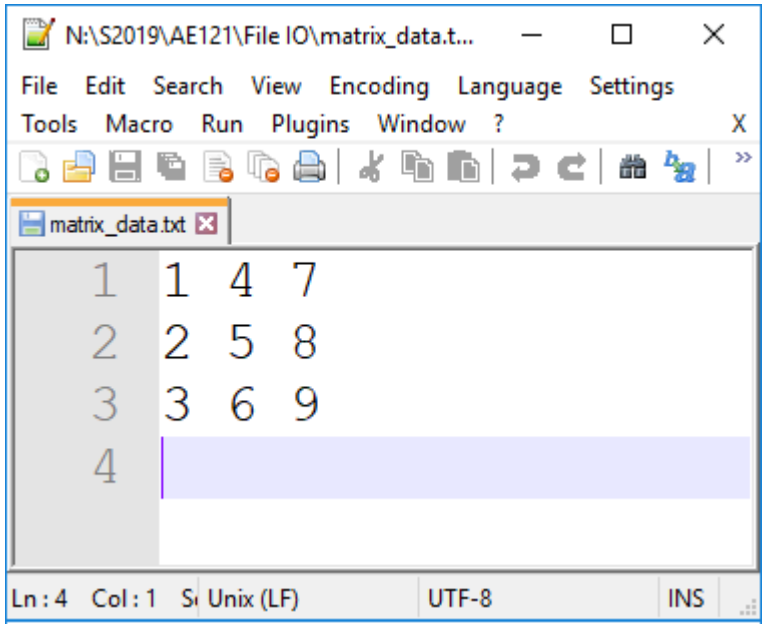
- If you want your matrix to appear as it does in your data, you must transpose your data

Example: Printing a Matrix of Values Without Transposing

Without transposing data

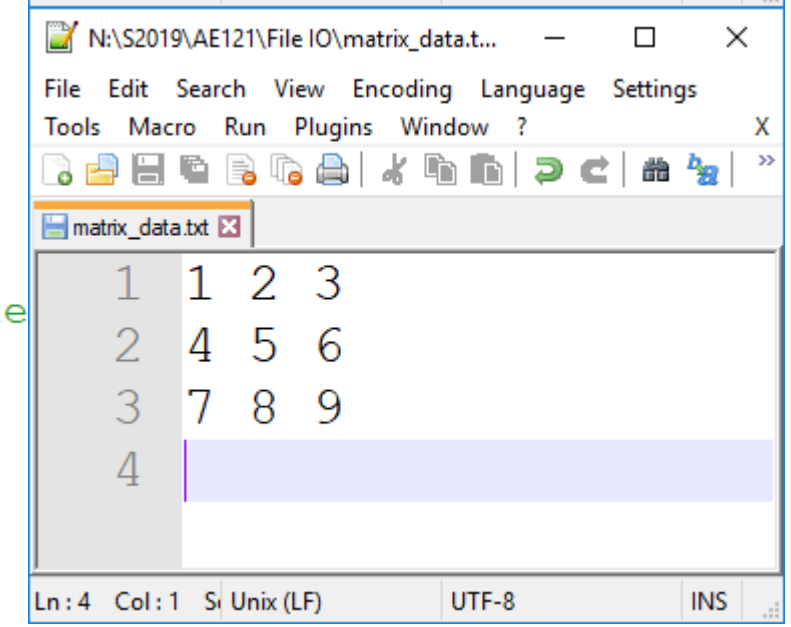
```
% Without transposing
mat_data = [1 2 3;4 5 6;7 8 9];
fid = fopen('matrix_data.txt', 'w');
fprintf(fid, '%d %d %d\n', mat_data); % Print to text file
fclose(fid);
```

```
mat_data =

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



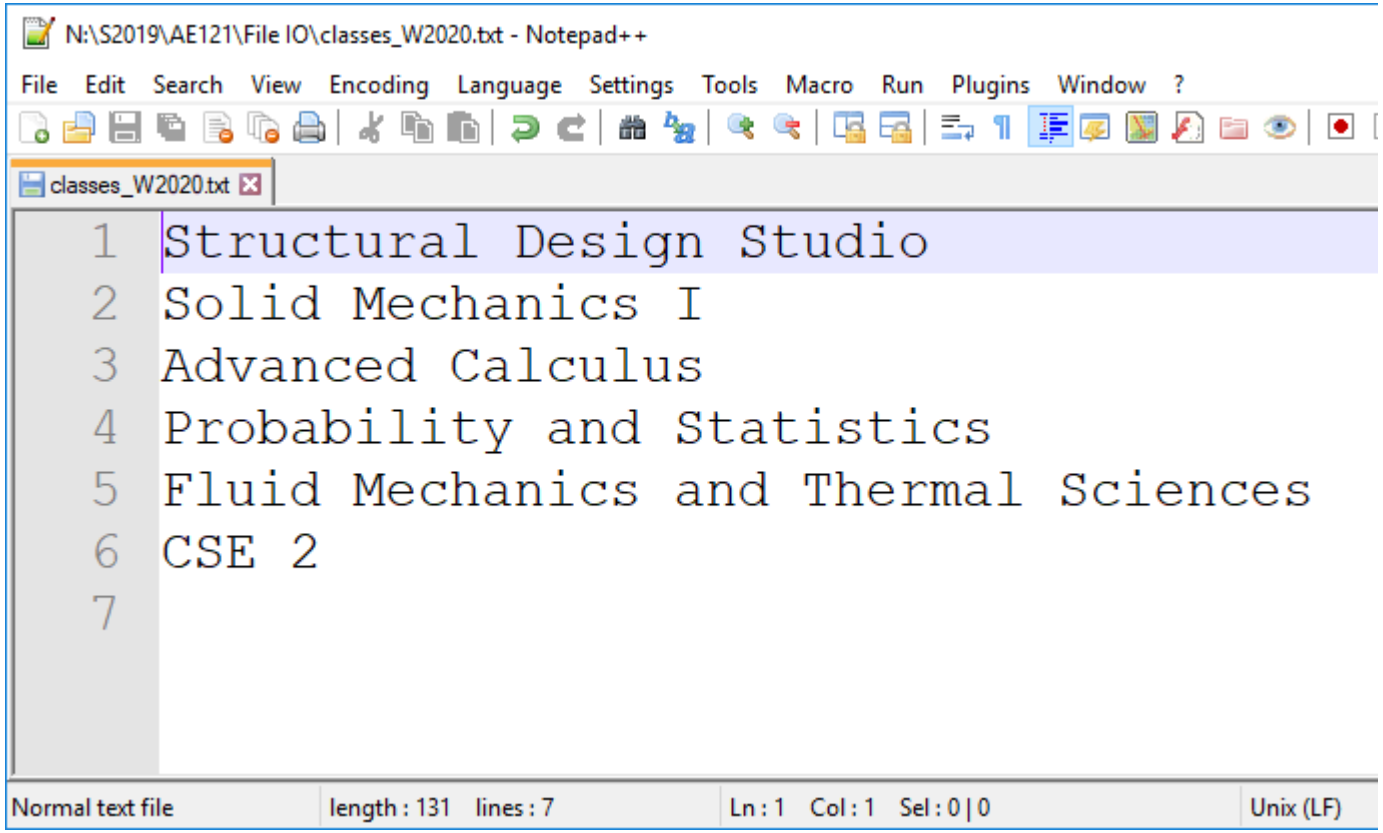
Transposing data

```
% Transposing
mat_data = [1 2 3;4 5 6;7 8 9];
fid = fopen('matrix_data.txt', 'w');
fprintf(fid, '%d %d %d\n', mat_data'); % Print to text file
fclose(fid);
```



Example: Text Data

```
courses_2A = ["Structural Design Studio"; "Solid Mechanics I"; ...
    "Advanced Calculus"; "Probability and Statistics"; ...
    "Fluid Mechanics and Thermal Sciences"; "CSE 2"];

fid = fopen('classes_W2020.txt', 'w');

fprintf(fid, '%s\n', courses_2A); % Print to text file

fclose(fid);
```



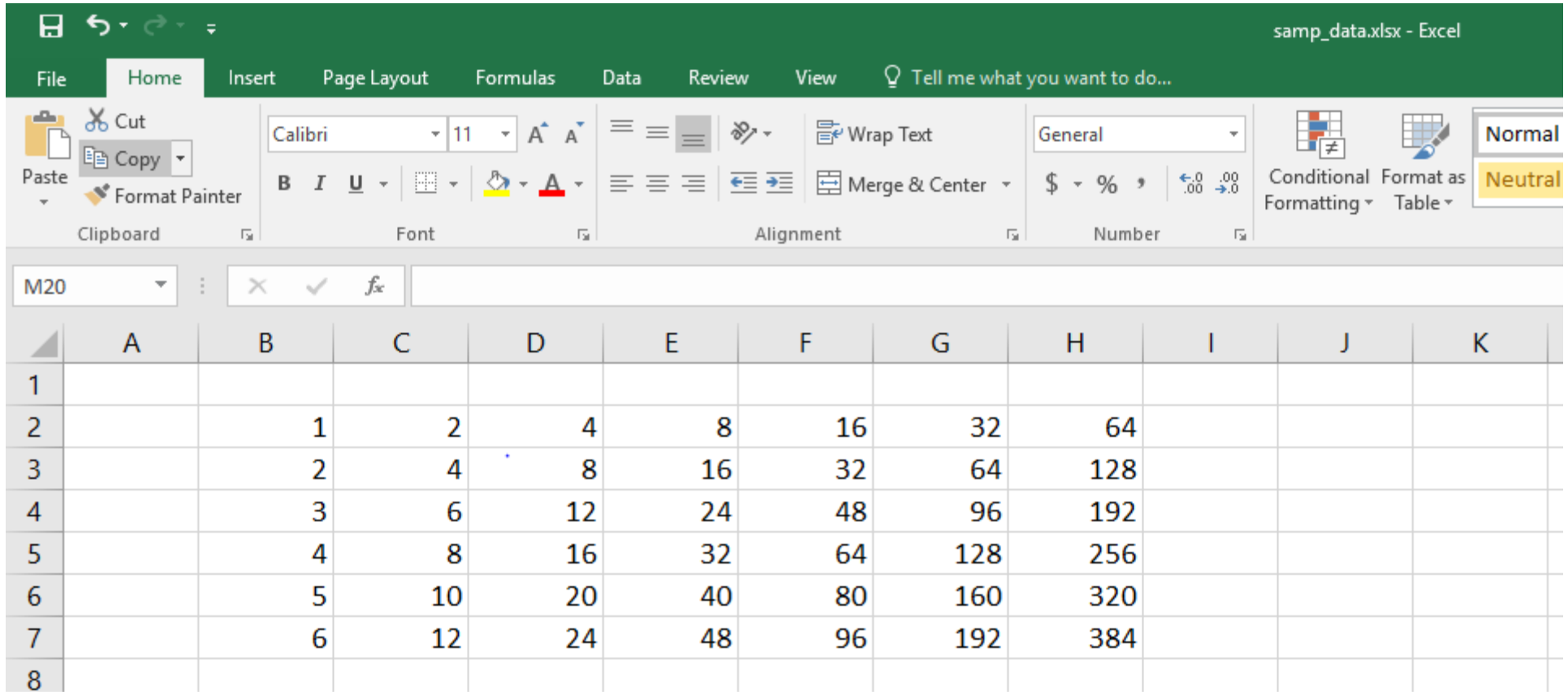
Reading Excel Data

The xlsread function can be used for importing Excel data into MATLAB.

For numeric data:
data = xlsread(filename, sheetname);
Ex. num_data = xlsread(‘file_01.xlsx’, ‘Force Calculations’)

For mixed text and numeric data, it is often easiest to download it as raw:
[num, txt, **raw**] = xlsread(filename, sheetname)
Ex. [~, ~, all_data] = xlsread(WheatonAve_project.xlsx’, ‘Sheet 1’)

Example: Reading Numeric Data



	A	B	C	D	E	F	G	H	I	J	K
1											
2		1	2	4	8	16	32	64			
3		2	4	8	16	32	64	128			
4		3	6	12	24	48	96	192			
5		4	8	16	32	64	128	256			
6		5	10	20	40	80	160	320			
7		6	12	24	48	96	192	384			
8											

Numeric Data Example

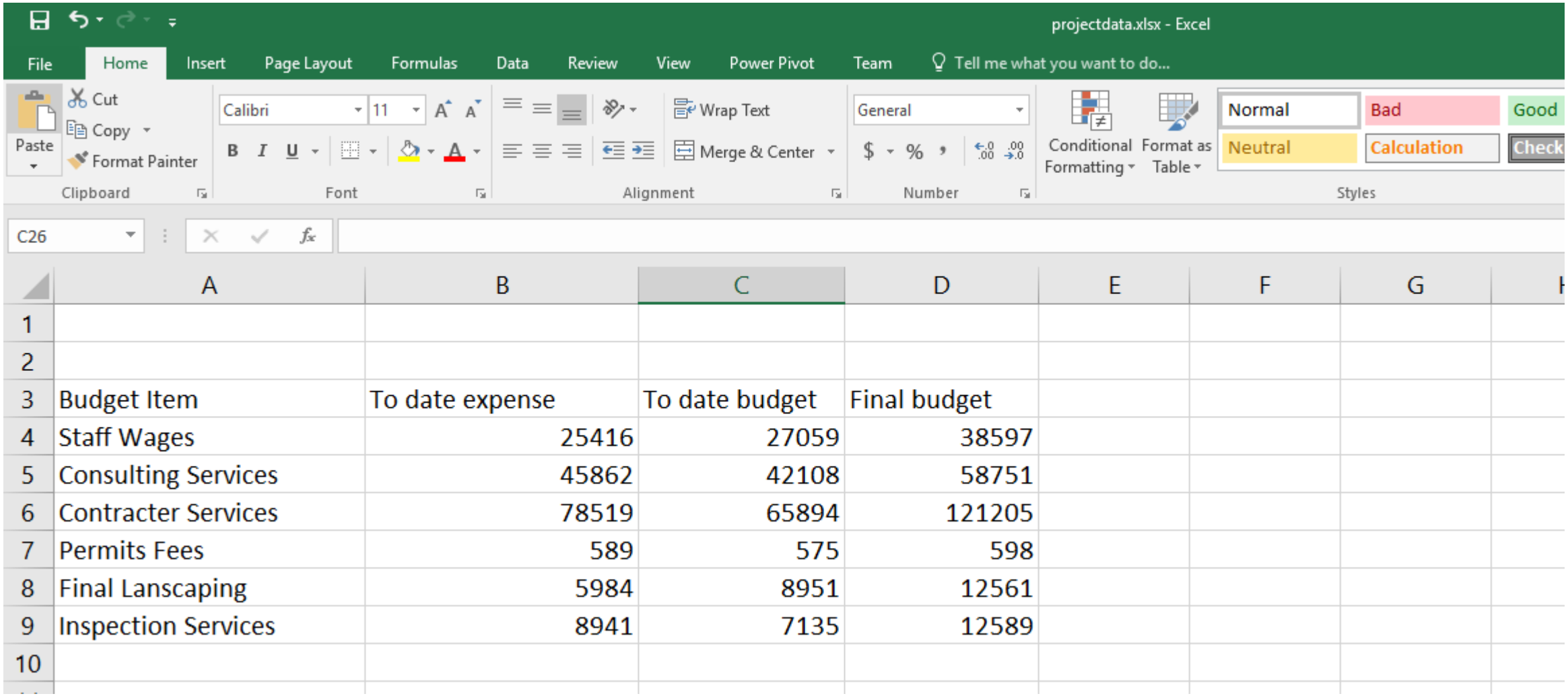
```
num_data = xlsread('samp_data.xlsx', 'Numeric Data')
```

num_data = 6×7

1	2	4	8	16	32	64
2	4	8	16	32	64	128
3	6	12	24	48	96	192
4	8	16	32	64	128	256
5	10	20	40	80	160	320
6	12	24	48	96	192	384

Example from Lab 09: Reading Mixed Data

Data from 'Project Budget' Excel spreadsheet



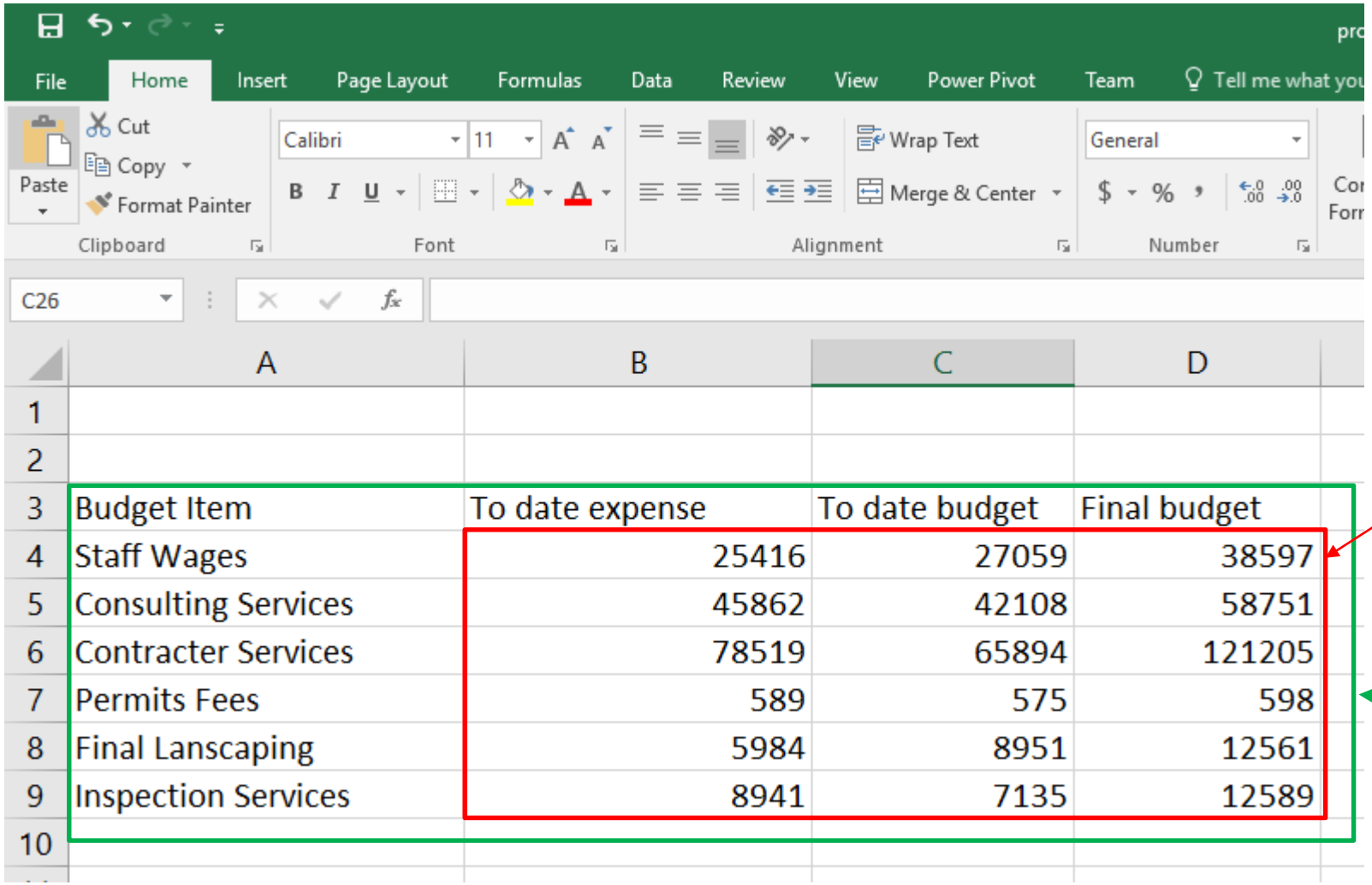
	A	B	C	D
1				
2				
3	Budget Item	To date expense	To date budget	Final budget
4	Staff Wages	25416	27059	38597
5	Consulting Services	45862	42108	58751
6	Contracter Services	78519	65894	121205
7	Permits Fees	589	575	598
8	Final Lanscaping	5984	8951	12561
9	Inspection Services	8941	7135	12589
10				

Mixed Data Type Example

```
[num,~,~] = xlsread('projectdata.xlsx', 'Project Budget')
[~, txt, ~] = xlsread('projectdata.xlsx', 'Project Budget')
[~,~,all_data] = xlsread('projectdata.xlsx', 'Project Budget')
```

```
num = 6×3
        25416       27059       38597
        45862       42108       58751
        78519       65894      121205
          589         575         598
         5984        8951       12561
         8941        7135       12589

txt = 7×4 cell array
    {'Budget Item'        }    {'To date expense'}    {'To date budget'}    {'Final budget'}
    {'Staff Wages'        }    {0×0 char         }    {0×0 char        }    {0×0 char      }
    {'Consulting Services'}    {0×0 char         }    {0×0 char        }    {0×0 char      }
    {'Contracter Services'}    {0×0 char         }    {0×0 char        }    {0×0 char      }
    {'Permits Fees'       }    {0×0 char         }    {0×0 char        }    {0×0 char      }
    {'Final Lanscaping'   }    {0×0 char         }    {0×0 char        }    {0×0 char      }
    {'Inspection Services'}    {0×0 char         }    {0×0 char        }    {0×0 char      }
all_data = 7×4 cell array
    {'Budget Item'        }    {'To date expense'}    {'To date budget'}    {'Final budget'}
    {'Staff Wages'        }    {[          25416]}    {[         27059]}    {[       38597]}
    {'Consulting Services'}    {[          45862]}    {[         42108]}    {[       58751]}
    {'Contracter Services'}    {[          78519]}    {[         65894]}    {[      121205]}
    {'Permits Fees'       }    {[            589]}    {[           575]}    {[         598]}
    {'Final Lanscaping'   }    {[           5984]}    {[          8951]}    {[       12561]}
    {'Inspection Services'}    {[           8941]}    {[          7135]}    {[       12589]}
```

Mixed Data Type Example

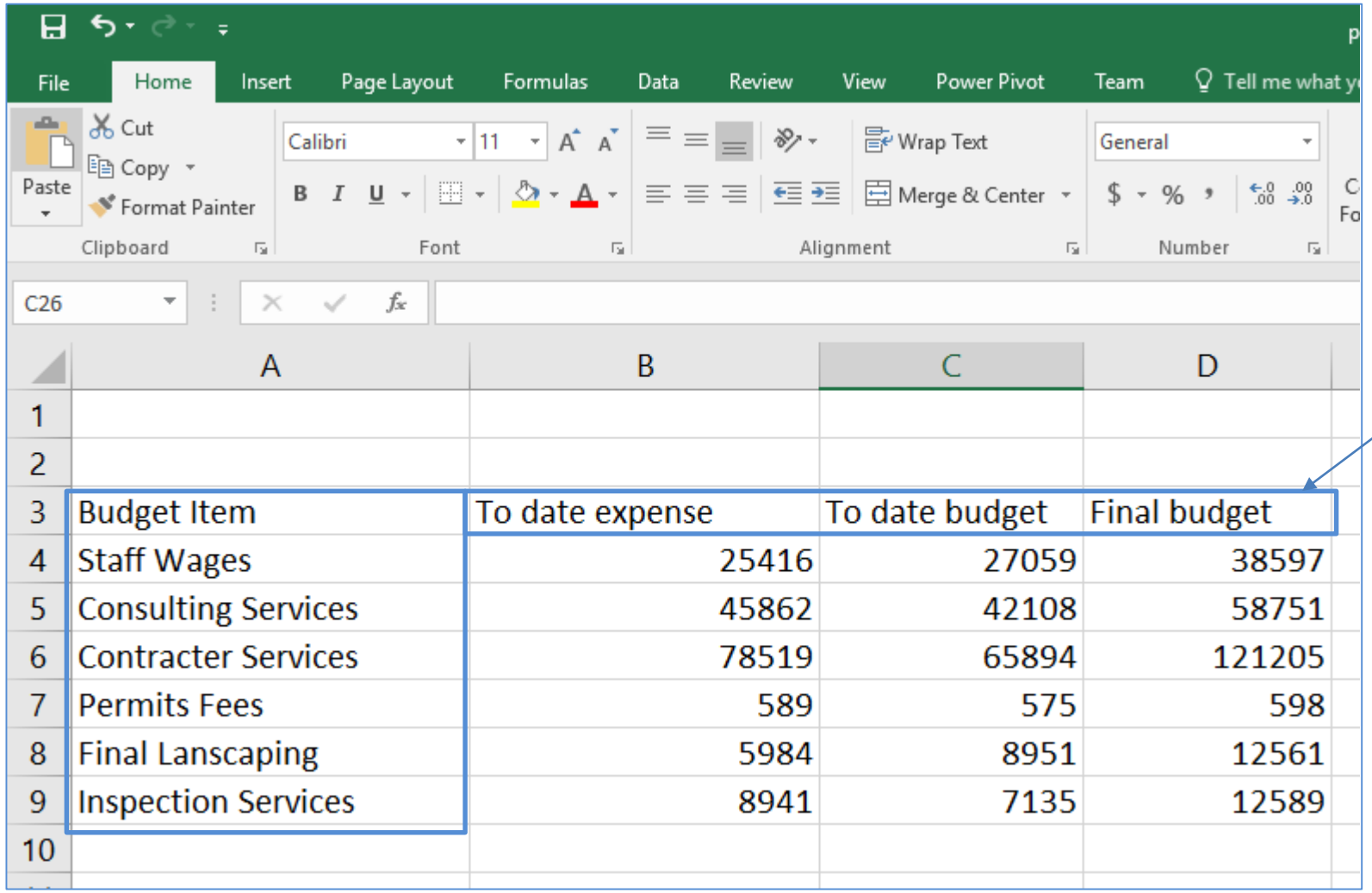


	A	B	C	D
1				
2				
3	Budget Item	To date expense	To date budget	Final budget
4	Staff Wages	25416	27059	38597
5	Consulting Services	45862	42108	58751
6	Contracter Services	78519	65894	121205
7	Permits Fees	589	575	598
8	Final Lanscaping	5984	8951	12561
9	Inspection Services	8941	7135	12589
10				

Numeric Data

All data (raw)

Mixed Data Type Example



	A	B	C	D
1				
2				
3	Budget Item	To date expense	To date budget	Final budget
4	Staff Wages	25416	27059	38597
5	Consulting Services	45862	42108	58751
6	Contracter Services	78519	65894	121205
7	Permits Fees	589	575	598
8	Final Lanscaping	5984	8951	12561
9	Inspection Services	8941	7135	12589
10				

Text data

Writing Excel Data

Writing Excel Data can be done using the **xlswrite** function

To write to a file in no specific location in the Excel file,

xlswrite(filename, data)

To specify what cells you want written to,

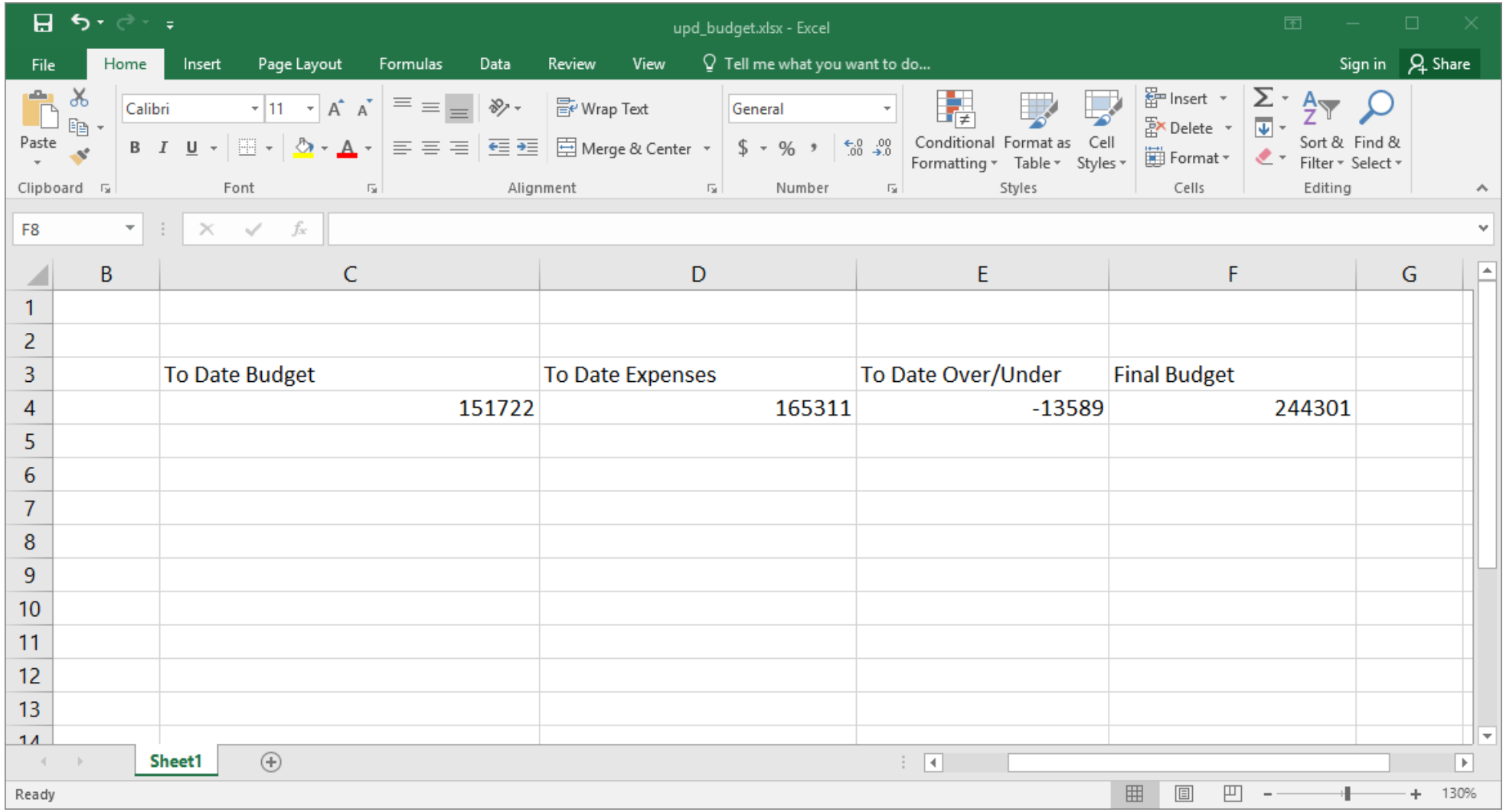
xlswrite(filename, data, ‘cell1:cell2’)

Filename should be in quotation marks
Ex. ‘file1.xlsx’

‘cell1’ is top left cell you want your data to cover and ‘cell2’ is the bottom right cell you want your data to cover

Example from Lab 09: Writing Tabular Data

Question: Write a row of text data and beneath it a row of numeric data to Excel in the locations shown.



Example from Lab 09: Tips

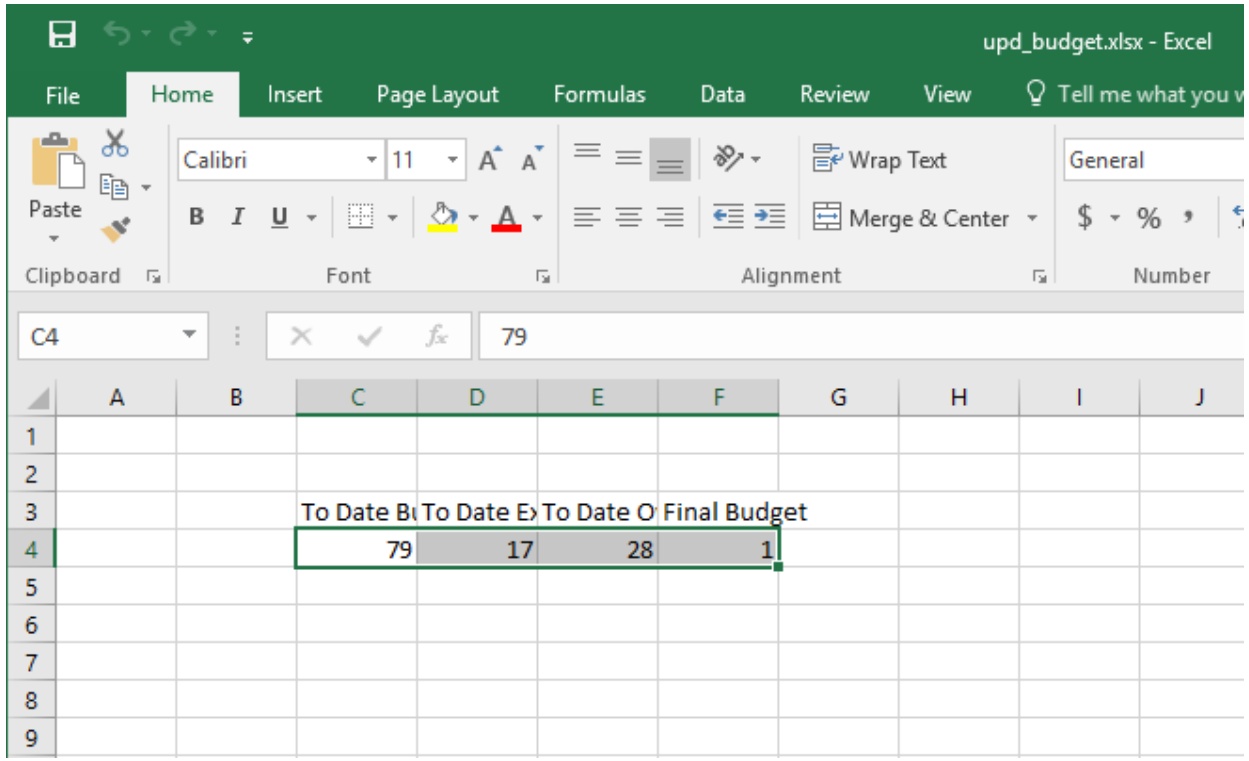
If you want to insert a row of data, make sure your data is a **row vector**

You can write on a file multiple times using xlswrite

Example from Lab 09

```
% (f)
titles = ["To Date Budget" "To Date Expenses" "To Date Over/Under" ...
    "Final Budget"];
data = randi(100,1,5); % Not the actual data for the question

xlswrite('upd_budget.xlsx', titles, 'C3:F3');
xlswrite('upd_budget.xlsx', data, 'C4:F4');
```

File will appear in the directory you are currently working in



MATLAB 2019 Updates

- MATLAB has new functions to work with Excel files, and others

- Some of these functions are:
  - readtable
  - readmatrix
  - writetable
  - writematrix