Data Import and Export

Data access from Text files, spreadsheets, images;

File I/O functions – read, write, delete

https://in.mathworks.com/help/matlab/import_export/supported-file-formats-for-import-and-export.html

Text files

- Read and write numeric and non-numeric data in delimited and formatted text files like **.csv** and **.txt** files.

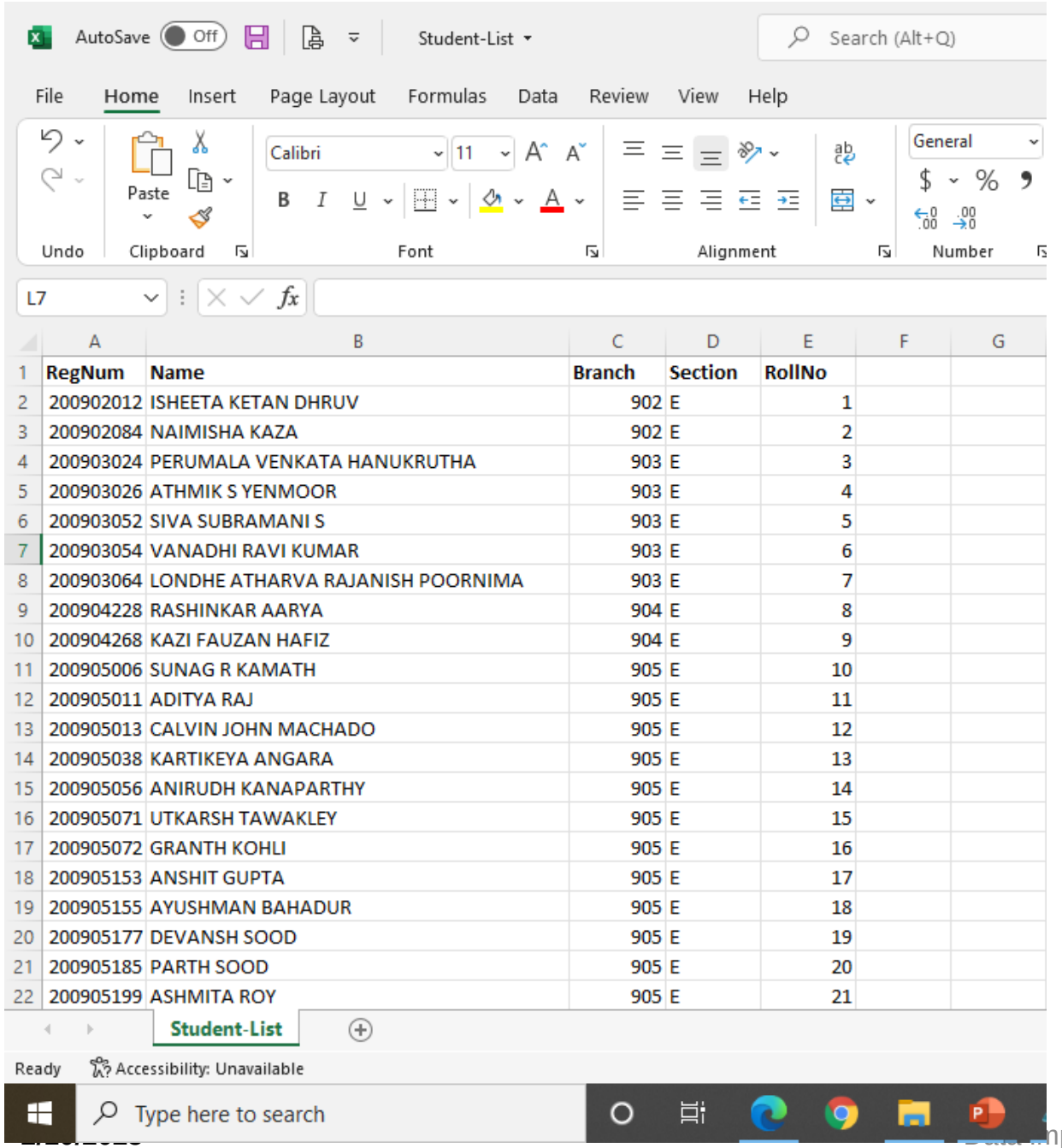
- **Text** files often contain a **mix** of **numeric** and **text** data as well as **variable** and **row** names.

- You can represent this data in MATLAB as **tables**, **timetables**, **matrices**, **cell arrays**, or **string arrays**.

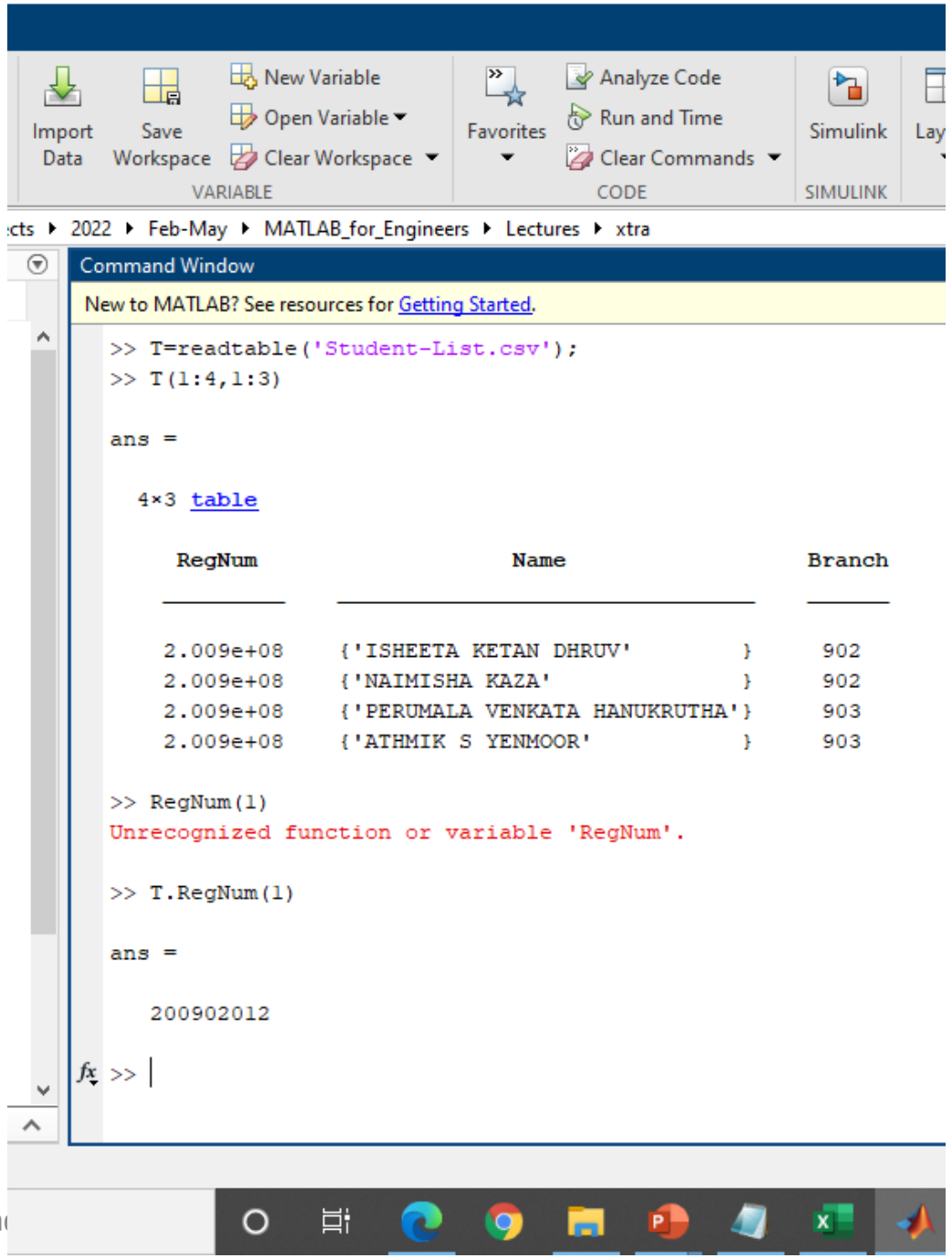
Import Data as Tables

- If text file has tabular data, it can be imported as a table.
- A table consists of column-oriented variables containing rows of data of the same type.
- Each variable in a table can hold a different data type and size, however, each variable must have the same number of rows.

- >> T = **readtable**(‘**Student-List**.**csv**’);
- >> T(1:5,1:5) **% Display the first five rows and columns of the table**.



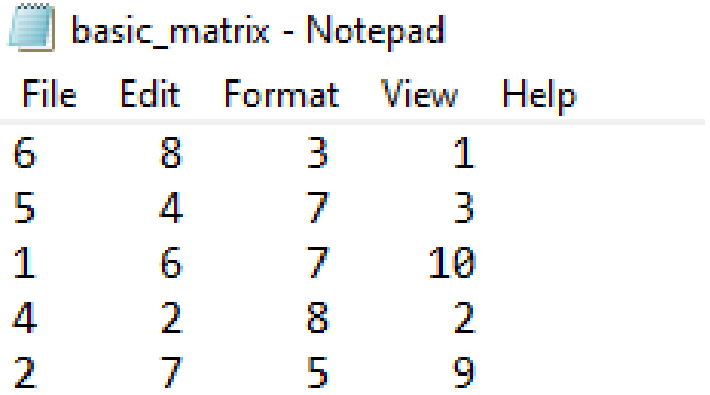
	A	B	C	D	E
1	RegNum	Name	Branch	Section	RollNo
2	200902012	ISHEETA KETAN DHRUV	902	E	1
3	200902084	NAIMISHA KAZA	902	E	2
4	200903024	PERUMALA VENKATA HANUKRUTHA	903	E	3
5	200903026	ATHMIK S YENMOOR	903	E	4
6	200903052	SIVA SUBRAMANI S	903	E	5
7	200903054	VANADHI RAVI KUMAR	903	E	6
8	200903064	LONDHE ATHARVA RAJANISH POORNIMA	903	E	7
9	200904228	RASHINKAR AARYA	904	E	8
10	200904268	KAZI FAUZAN HAFIZ	904	E	9
11	200905006	SUNAG R KAMATH	905	E	10
12	200905011	ADITYA RAJ	905	E	11
13	200905013	CALVIN JOHN MACHADO	905	E	12
14	200905038	KARTIKEYA ANGARA	905	E	13
15	200905056	ANIRUDH KANAPARTHY	905	E	14
16	200905071	UTKARSH TAWAKLEY	905	E	15
17	200905072	GRANTH KOHLI	905	E	16
18	200905153	ANSHIT GUPTA	905	E	17
19	200905155	AYUSHMAN BAHADUR	905	E	18
20	200905177	DEVANSH SOOD	905	E	19
21	200905185	PARTH SOOD	905	E	20
22	200905199	ASHMITA ROY	905	E	21

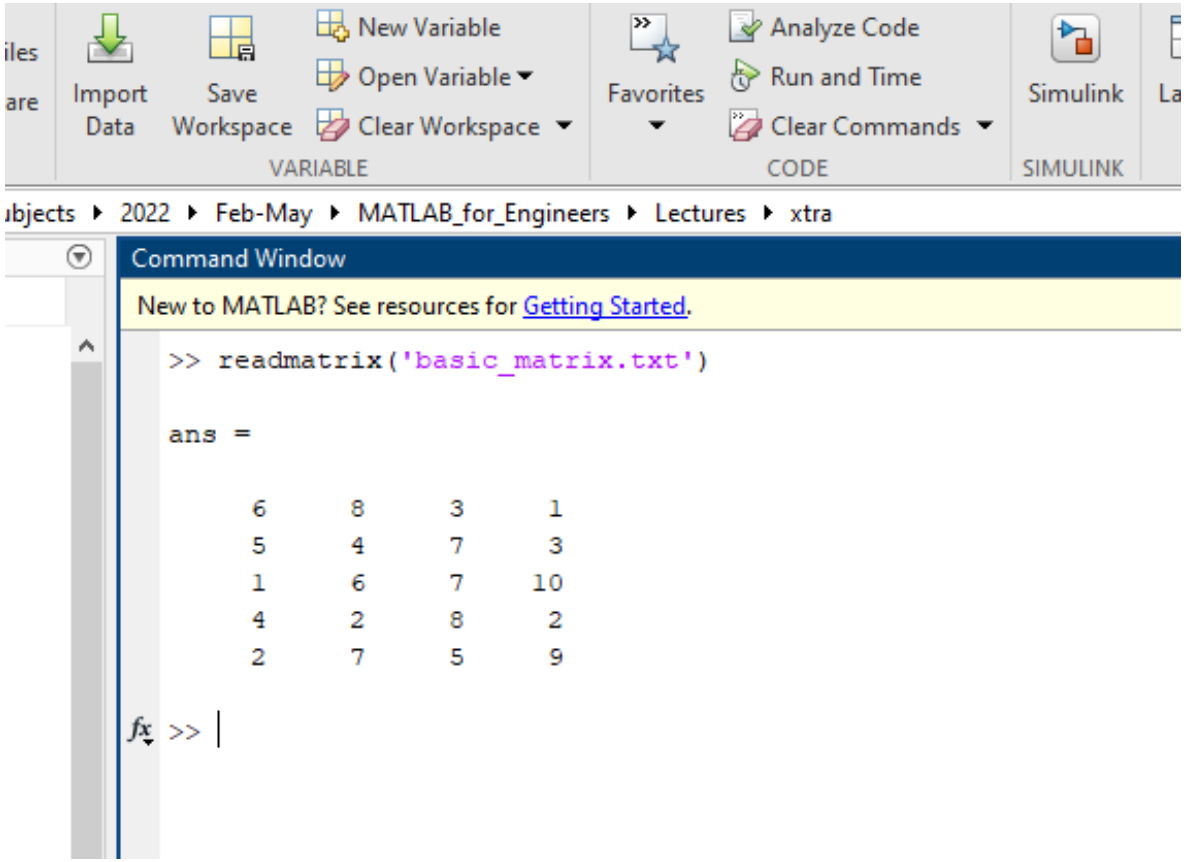


Import Data as Matrices

- If text file contains uniform data (all of the same type), it can be imported as a matrix. Importing data into a matrix allows to work with a minimally formatted array.

- >> M = readmatrix('basic_matrix.txt')





Import Data as Cell Arrays

- A cell array is a data type with **indexed data containers** called **cells**, where **each cell** can **contain any type of data**.

student-list - Notepad

File Edit Format View Help

```
200902012       ISHEETA KETAN DHRUV     902     E       1
200902084       NAIMISHA KAZA   902     E       2
200903024       PERUMALA VENKATA HANUKRUTHA     903     E       3
200903026       ATHMIK S YENMOOR        903     E       4
200903052       SIVA SUBRAMANI S        903     E       5
200903054       VANADHI RAVI KUMAR      903     E       6
200903064       LONDHE ATHARVA RAJANISH POORNIMA        903     E       7
200904228       RASHINKAR AARYA 904     E       8
200904268       KAZI FAUZAN HAFIZ       904     E       9
200905006       SUNAG R KAMATH  905     E       10
200905011       ADITYA RAJ      905     E       11
200905013       CALVIN JOHN MACHADO     905     E       12
200905038       KARTIKEYA ANGARA        905     E       13
200905056       ANIRUDH KANAPARTHY      905     E       14
200905071       UTKARSH TAWAKLEY        905     E       15
200905072       GRANTH KOHLI    905     E       16
200905153       ANSHIT GUPTA    905     E       17
200905155       AYUSHMAN BAHADUR        905     E       18
200905177       DEVANSH SOOD    905     E       19
200905185       PARTH SOOD      905     E       20
200905199       ASHMITA ROY     905     E       21
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> C = readcell('student-list.txt')

C =

  21×5 cell array

    {[200902012]}    {'ISHEETA KETAN D…'}    {[902]}    {'E'}    {[ 1]}
    {[200902084]}    {'NAIMISHA KAZA'    }    {[902]}    {'E'}    {[ 2]}
    {[200903024]}    {'PERUMALA VENKAT…'}    {[903]}    {'E'}    {[ 3]}
    {[200903026]}    {'ATHMIK S YENMOOR' }    {[903]}    {'E'}    {[ 4]}
    {[200903052]}    {'SIVA SUBRAMANI S' }    {[903]}    {'E'}    {[ 5]}
    {[200903054]}    {'VANADHI RAVI KU…'}    {[903]}    {'E'}    {[ 6]}
    {[200903064]}    {'LONDHE ATHARVA …'}    {[903]}    {'E'}    {[ 7]}
    {[200904228]}    {'RASHINKAR AARYA'  }    {[904]}    {'E'}    {[ 8]}
    {[200904268]}    {'KAZI FAUZAN HAFIZ'}    {[904]}    {'E'}    {[ 9]}
    {[200905006]}    {'SUNAG R KAMATH'   }    {[905]}    {'E'}    {[10]}
    {[200905011]}    {'ADITYA RAJ'       }    {[905]}    {'E'}    {[11]}
    {[200905013]}    {'CALVIN JOHN MAC…'}    {[905]}    {'E'}    {[12]}
    {[200905038]}    {'KARTIKEYA ANGARA' }    {[905]}    {'E'}    {[13]}
    {[200905056]}    {'ANIRUDH KANAPAR…'}    {[905]}    {'E'}    {[14]}
    {[200905071]}    {'UTKARSH TAWAKLEY' }    {[905]}    {'E'}    {[15]}
    {[200905072]}    {'GRANTH KOHLI'     }    {[905]}    {'E'}    {[16]}
    {[200905153]}    {'ANSHIT GUPTA'     }    {[905]}    {'E'}    {[17]}
    {[200905155]}    {'AYUSHMAN BAHADUR' }    {[905]}    {'E'}    {[18]}
    {[200905177]}    {'DEVANSH SOOD'     }    {[905]}    {'E'}    {[19]}
    {[200905185]}    {'PARTH SOOD'       }    {[905]}    {'E'}    {[20]}
    {[200905199]}    {'ASHMITA ROY'      }    {[905]}    {'E'}    {[21]}
```

textscan

- Read formatted data from text file or string

- **fileID** = **fopen**(‘student-list.txt’,'r');
- C = textscan(**fileID**,’%s %s %s %s %s','Delimiter',’ ');
- fclose(fileID);

- Format Specifiers :
- https://in.mathworks.com/help/matlab/ref/textscan.html

MATLAB R2021a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

Search Documentation | Shailesh

New Open Save Find Files Compare Print — FILE
Go To Find — NAVIGATE
Insert Comment Indent — EDIT
Breakpoints — BREAKPOINTS
Run Run and Advance Run Section Advance Run and Time — RUN

E: ▸ office ▸ subjects ▸ 2022 ▸ Feb-May ▸ MATLAB_for_Engineers ▸ Lectures ▸ xtra

Editor - E:\office\subjects\2022\Feb-May\MATLAB_for_Engineers\Lectures\xtra\read4mtxtfile.m

read4mtxtfile.m

```
1 -    clear all;clc;
2 -    fileID = fopen('student-list.txt','r');
3 -    C = textscan(fileID,'%s %s %s %s %s','Delimiter','');
4 -    fclose(fileID);
5
```

Current Folder

Name

- tutorial_01.mlx
- Tut_03_while.m
- Tut_03_binary_search.m
- Tut_03_arr_rev_itr.m
- Tut_03_03.m
- Tut_03_02.m
- Tut_03_01.m
- Tut_02_06.m
- Tut_02_03.mlx
- tut_02_03.m
- tut_02_02.m
- tut_02_01.m
- test_live.mlx
- test.m
- T02-Live_script.mlx
- Student-List.csv
- student-list.txt
- recArrayRev.m

Details

Command Window

New to MATLAB? See resources for Getting Started.

```
>> celldisp(C)

C{1}{1} =

200902012   ISHEETA KETAN DHRUV 902 E   1


C{1}{2} =

200902084   NAIMISHA KAZA   902 E   2


C{1}{3} =

200903024   PERUMALA VENKATA HANUKRUTHA 903 E   3


C{1}{4} =
```

Workspace

Name	Value
ans	0
C	1x5 cell
fileID	3

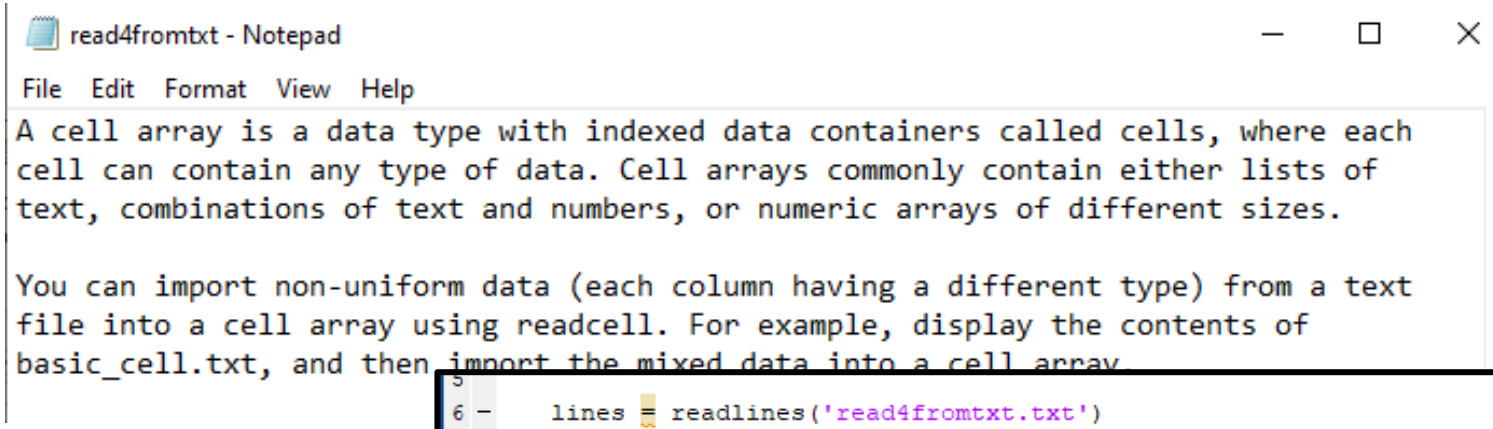
Type here to search

28°C Mostly cloudy

09:11 AM
26-Mar-22

Import Data as String Arrays

- If text file contains lines of plain text, it can be represented as plain text in MATLAB as a string array.




```
6 -    lines = readlines('read4fromtxt.txt')
```

Current Folder

Name

- tutorial_01.mlx
- Tut_03_while.m
- Tut_03_binary_search.m
- Tut_03_arr_rev_itr.m
- Tut_03_03.m
- Tut_03_02.m
- Tut_03_01.m
- Tut_02_06.m
- Tut_02_03.mlx
- tut_02_03.m
- tut_02_02.m
- tut_02_01.m

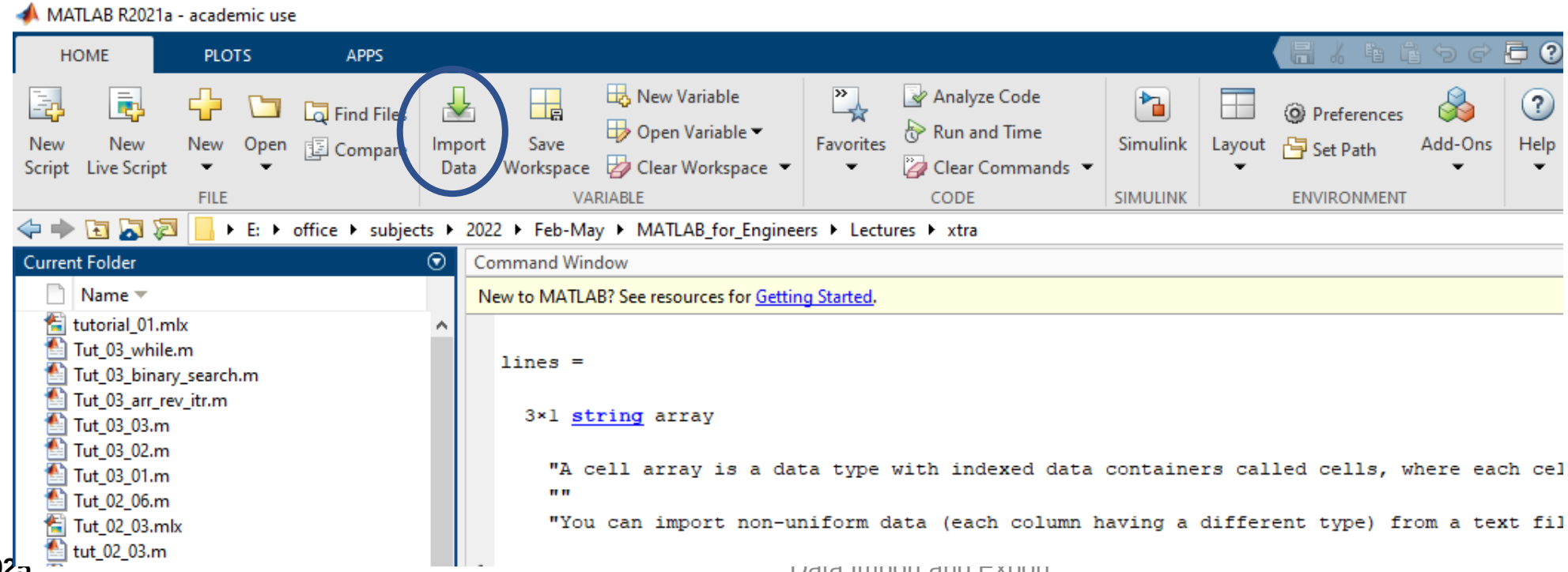
Command Window

New to MATLAB? See resources for Getting Started.

```
lines =

  3×1 string array

    "A cell array is a data type with indexed data containers called cells, where each cell ca
    ""
    "You can import non-uniform data (each column having a different type) from a text file ir

fx >>
```

Import Tool (Homework / Self study)

- The Import Tool lets you preview and import data from spreadsheet files, delimited text files, and fixed-width text files.
- You can interactively select the data to import and reuse the script or function that the tool generates to import other similar files.



Importing Spreadsheets (Excel files)

- Import Spreadsheet Data Using **readtable**
  - T = **readtable**('patients.xls’);
  - T = **readtable**('patients.xls','Range','A1:E5’)
  - Specify the range in Excel notation as 'A1:E5' to read the first five rows and columns of the spreadsheet.

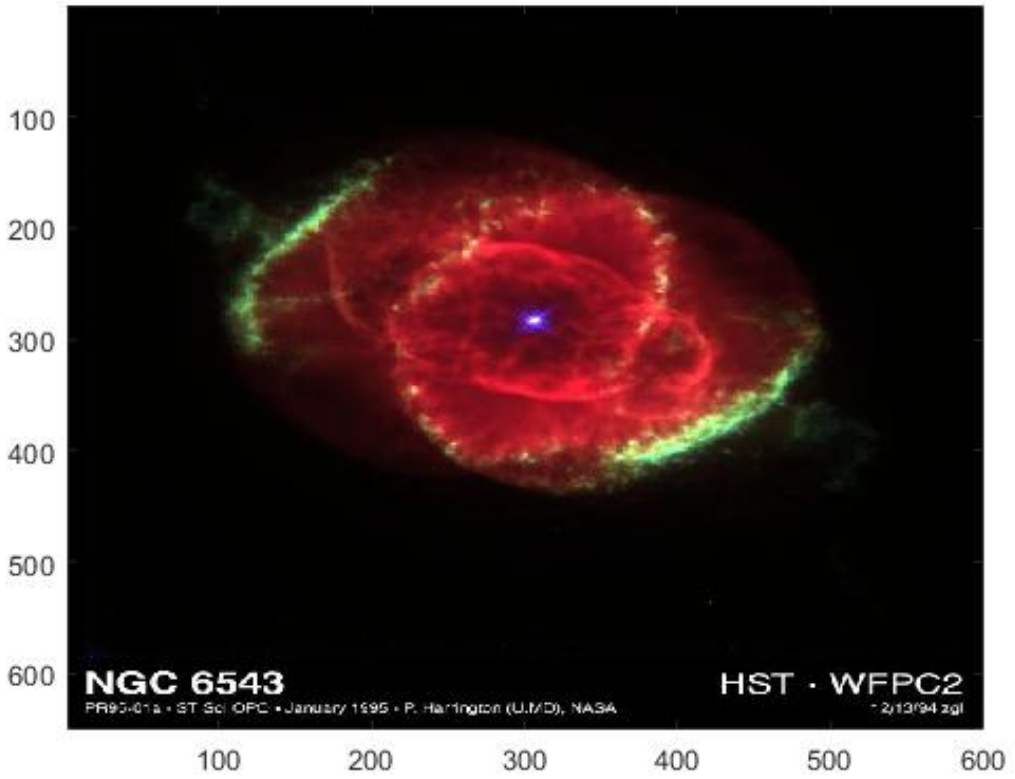
- Read Spreadsheet Data into Matrix
  - M = **readmatrix**('basic_matrix.xls')
  - M = **readmatrix**('basic_matrix.xls','Sheet','Sheet1','Range','B1:D3')

Importing Images

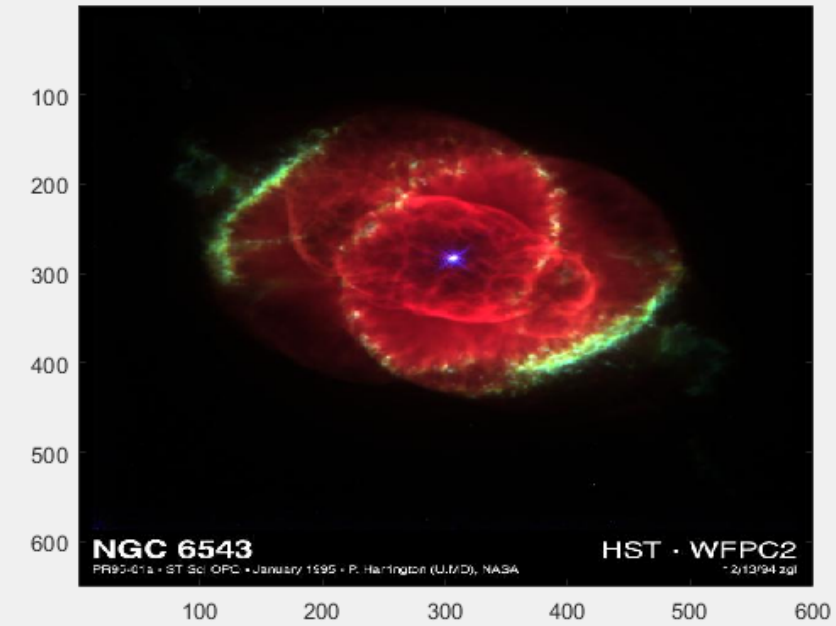
- To import data into the MATLAB workspace from a graphics file, use the **imread** function.

- >> A = **imread**('ngc6543a.jpg');
- >> image(A); % Display image file

- >> info = **imfinfo**(filename) –
- Information about graphics file



```
>> A = imread('ngc6543a.jpg');
>> image(A)
>> imfinfo('ngc6543a.jpg')

ans =

  struct with fields:

          Filename: 'E:\MATLAB\toolbox\matlab\demos
      FileModDate: '02-Oct-1996 01:49:44'
         FileSize: 27387
           Format: 'jpg'
    FormatVersion: ''
            Width: 600
           Height: 650
         BitDepth: 24
        ColorType: 'truecolor'
  FormatSignature: ''
  NumberOfSamples: 3
     CodingMethod: 'Huffman'
    CodingProcess: 'Sequential'
          Comment: {'CREATOR: XV Version 3.00b  Rev: 6/15/94  Quality = 75, Smoothing = 0

>> image(A)
>>
```



Low-Level File I/O

- **fscanf** - Read data from text file

Create a sample text file that contains floating-point numbers.

```
x = 100*rand(8,1);
fileID = fopen('nums1.txt','w');
fprintf(fileID,'%4.4f\n',x);
fclose(fileID);
```

```
>> writeandread
>> type 'nums1.txt'

81.4724
90.5792
12.6987
91.3376
63.2359
9.7540
27.8498
54.6882
>>
```

Reading from the txt file

Define the format of the data to read.
Use '%f' to specify floating-point numbers.

```
fileID = fopen('nums1.txt','r');
formatSpec = '%f';
A = fscanf(fileID,formatSpec);
fclose(fileID);
```

Export functions

Writing as TXT, XLS and IMAGEs

writematrix

- **writematrix**(A,**filename**) - writes to a file with the name and extension specified by **filename**.

- writematrix determines the file format based on the specified extension.
- The extension must be one of the following:
- **.txt, .dat, or .csv** for delimited text files
- **.xls, .xlsm, or .xlsx** for Excel spreadsheet files
- **.xlsb** for Excel spreadsheet files supported on systems with Excel for Windows

writematrix

- M = magic(5);

```
M = 5×5

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

Write the matrix to a comma delimited text file and display the file contents. The **writematrix** function outputs a text file named **M.txt**.

```
>>writematrix(M)
>>type 'M.txt'
```

To write the same matrix to a text file with a different delimiter character, use the 'Delimiter' name-value pair.

```
>>writematrix(M, 'M_tab.txt', 'Delimiter','tab')
>>type 'M_tab.txt'
```

Write the matrix to a spreadsheet file.

```
>>writematrix(M,'M.xls')
```

Read and display the matrix from M.xls.

```
>>readmatrix('M.xls')
```

Write the matrix to M.xls, to the second worksheet in the file, starting at the third row.

```
>>writematrix(M,'M.xls','Sheet',2,'Range','A3:E8')
```

Read and display the matrix.

```
>>readmatrix('M.xls','Sheet',2,'Range','A3:E8')
```

Append Data to Spreadsheet

- Append an array of data below existing data in a spreadsheet.

```
>> M1 = magic(5)  % First Matrix
>> writematrix(M1,'M.xls');
>> M2 = [5 10 15 20 25; 30 35 40 45 50]  % second matrix
>> writematrix(M2,'M.xls','WriteMode','append');
>> readmatrix('M.xls')
```

Append Matrix Data to Text File

- Append an array of data below existing data in a text file.

```
>> writematrix(M1,’M.txt’)
>> writematrix(M2,’M.txt','WriteMode','append’)
>> readmatrix('fibonacci.txt')
```

writetable

- Write table to file
- >> T = table(['M';'F';'M'],[45 45;41 32;40 34],{'NY';'CA';'MA'},[true; false; false])

```
T=3×4 table
    Var1      Var2      Var3      Var4
    ____    ________    ______    _____

     M        45        45      {'NY'}    true
     F        41        32      {'CA'}    false
     M        40        34      {'MA'}    false
```

Write the table to a **comma delimited text** file and display the file contents.

```
>>writetable(T)

writetable outputs a text file named T.txt.
>>type 'T.txt'
```

Write the table to a **space-delimited text** file named myData.txt and display the file contents.

```
>>writetable(T,'myData.txt','Delimiter',' ')
>>type 'myData.txt'
```

writetable

- Write the table to a **comma-separated text file** named **myData.csv** and view the file contents. Use the **'QuoteStrings**' name-value pair argument to **ensure** that the **commas** in the **third column are not treated as delimiters**.

- >> **writetable**(T,'myData.csv','Delimiter',',','QuoteStrings',true)
- >> **type** 'myData.csv'

writetable

- Write Table to Specific Sheet and Range in Spreadsheet

- Write the table to a spreadsheet named myData.xls. Include the data on the first sheet in the 5-by-5 region with corners at B2 and F6. You can change the worksheet to write to by specifying the index corresponding to the worksheet.

- >> **writetable**(T,'myData.xls','Sheet',1,'Range','B2:F6')

writelines

- Write text to file

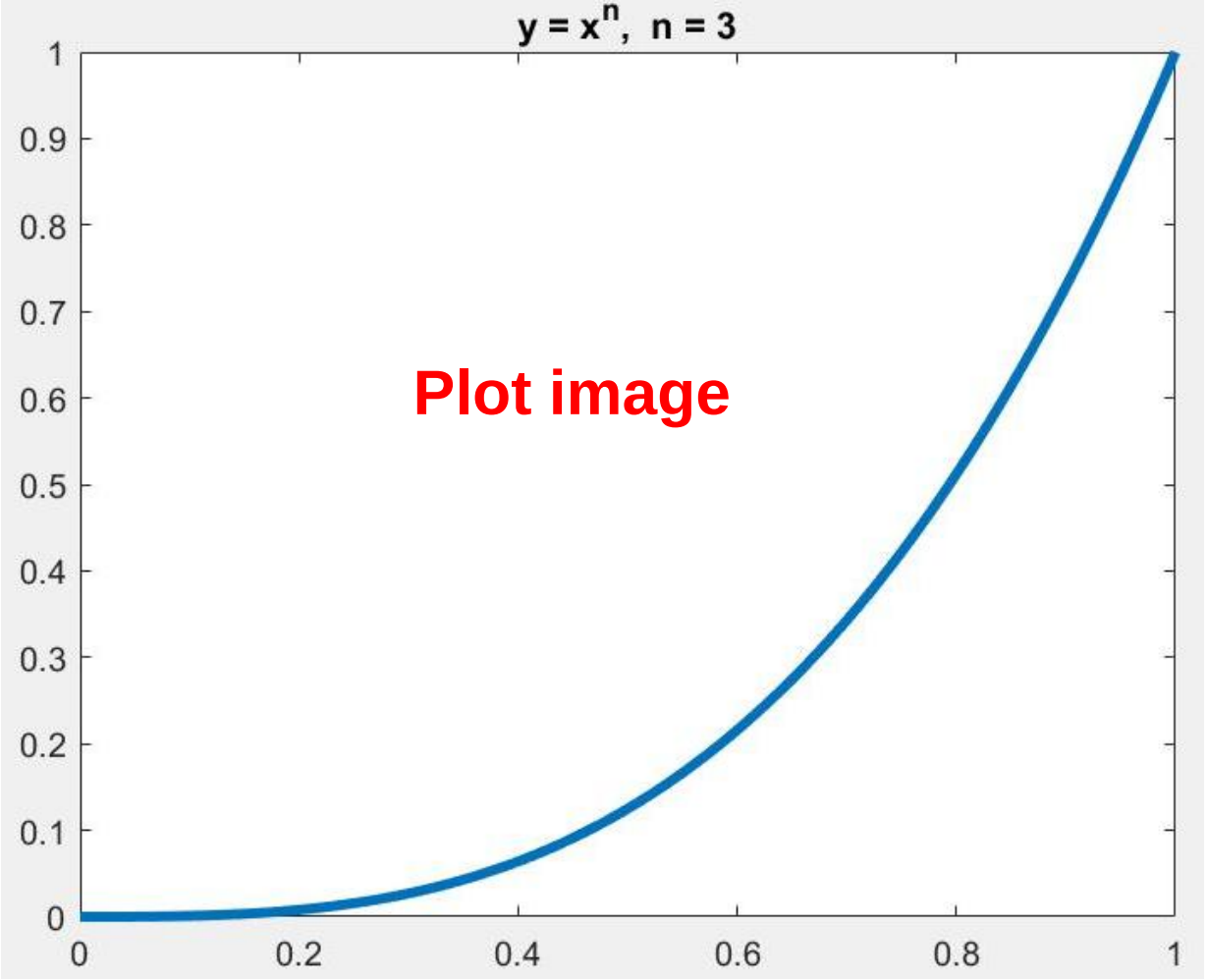
- Write the text "Example String" to a new file within the current directory.
- >>**writelines**("Example String","temp.txt")
- Display the contents of the new file.
- >>**type** temp.txt

- Append a string to an existing file.
- **lines** = "New Content 456";
- **filename** = "temp.txt";
- **writelines**(**lines**,filename,WriteMode="append")

imwrite

- Write image to graphics file

```
x = 0:0.01:1;
n = 3;
y = x.^n;
plot(x,y,'LineWidth',3);
title(['y = x^n,  n = ' num2str(n) ]);

H = getframe(gcf);

% save the image:
imwrite(H.cdata, 'testimage.jpg');
```



Delete file

- Delete files from CURRENT directory
- >>**delete** filename1 … Filenamen
- To delete all files in the current folder with a **.mat** extension.
- **>> delete *.mat**