Machine Learning

tanshuqiu

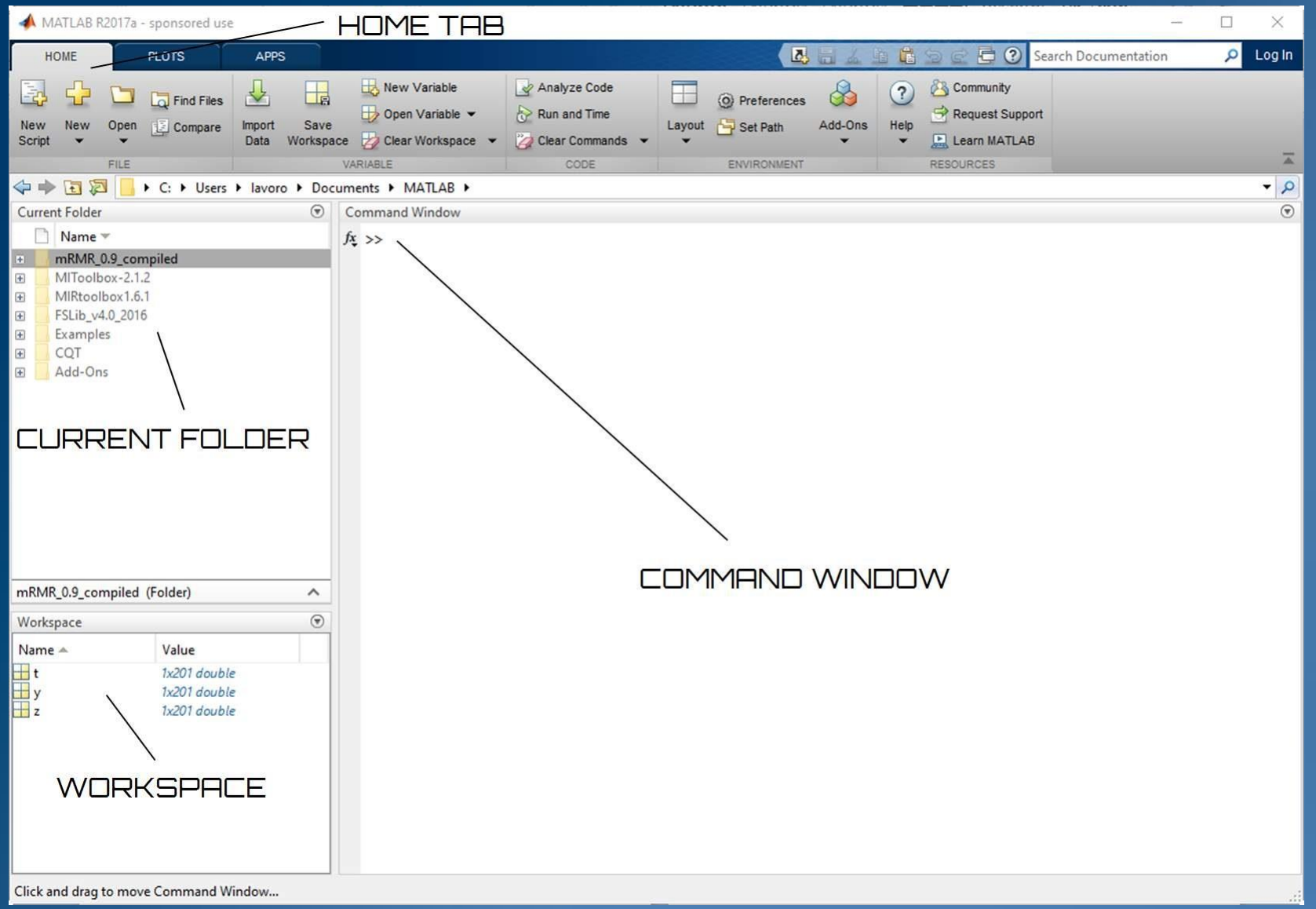
Email: tsq@cqut.edu.cn

Importing and Organizing Data in MATLAB

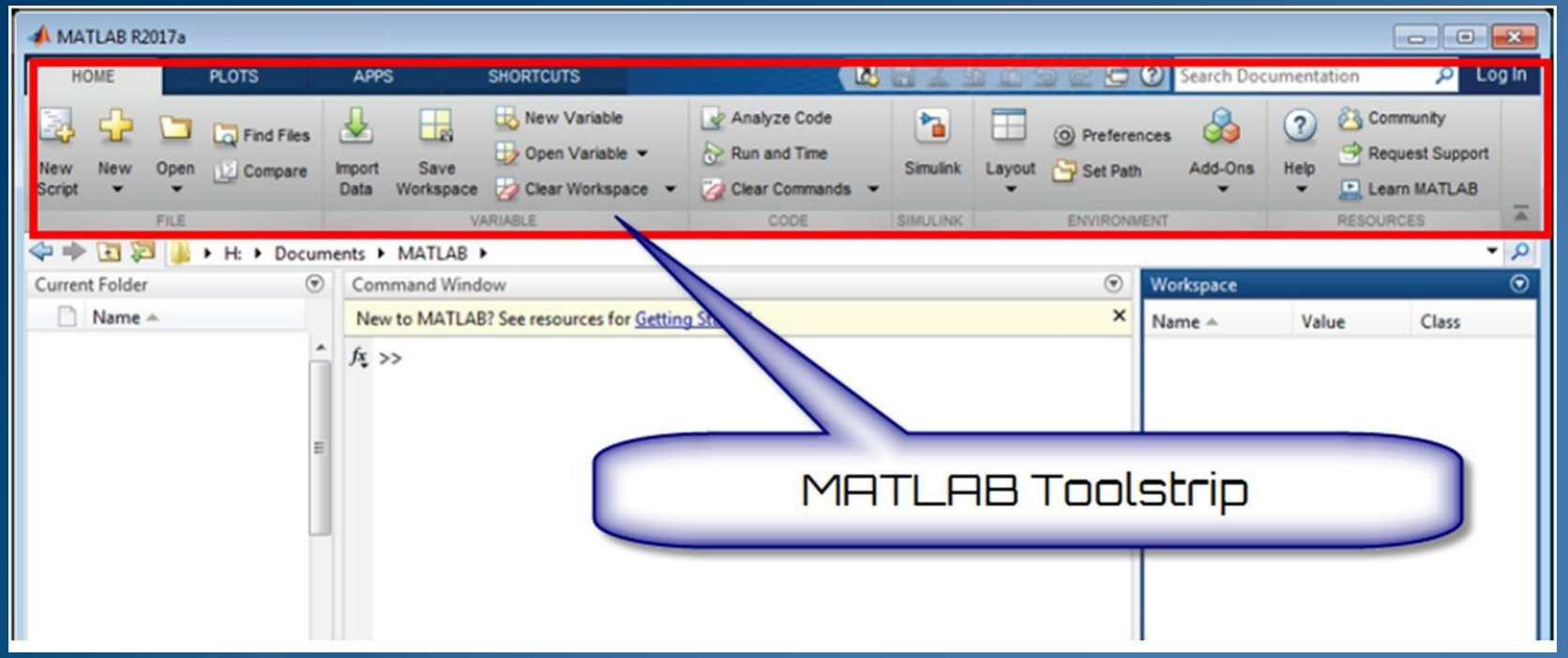
1. Familiarizing yourself with the MATLAB desktop
2. Importing data into MATLAB
3. Exporting data from MATLAB
4. Working with media files
5. Data organization

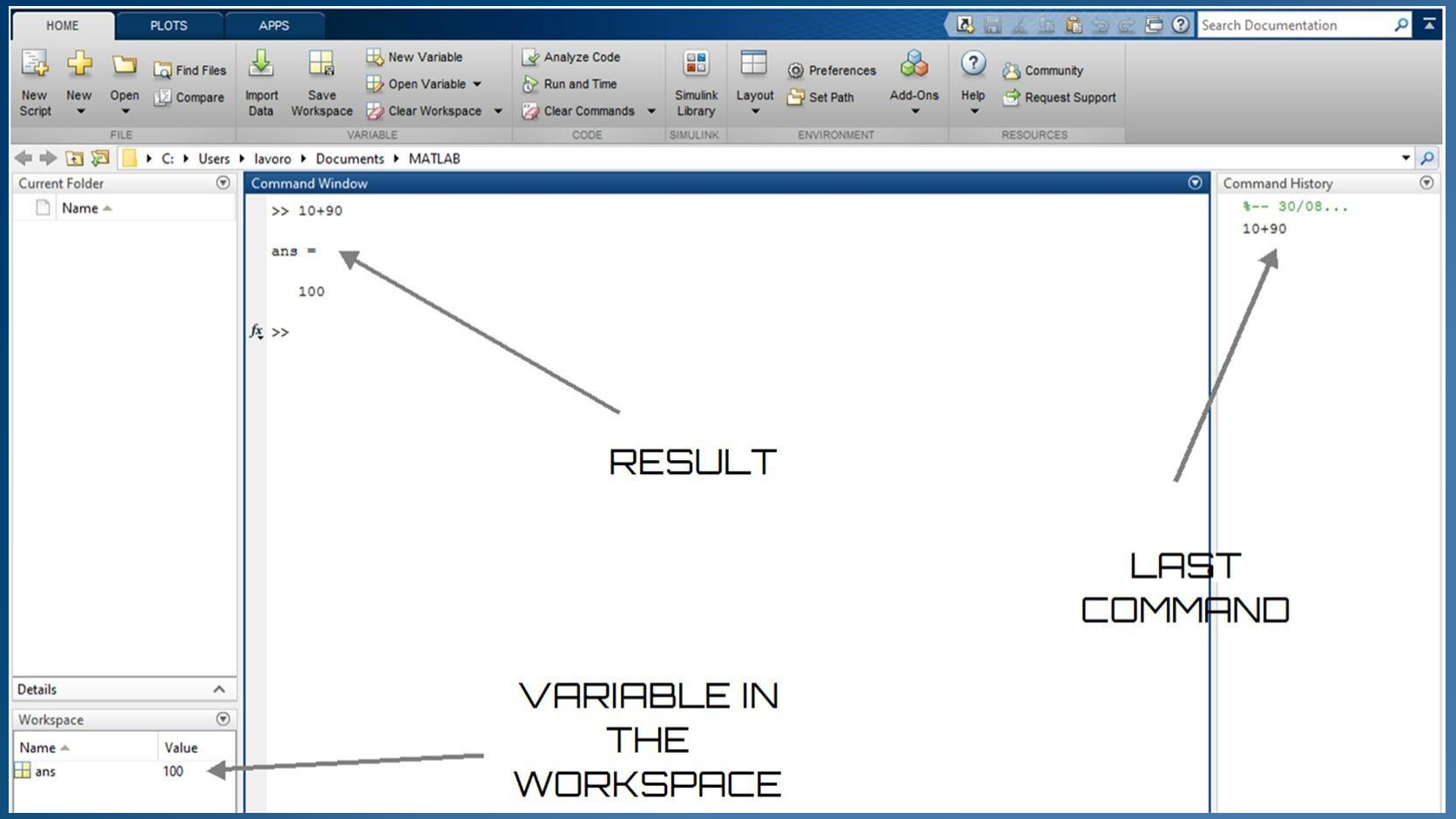
1. Familiarizing yourself with the MATLAB desktop

MATLAB is an interactive working environment based on the matrix, the most natural way to express computational mathematics. Moreover, it is a programming language designed for technical computing, mathematical analysis, and system simulation.



MATLAB R2017a desktop for Windows 10


MATLAB R2017a
HOME
PLOTS
APPS
SHORTCUTS
Search Documentation
Log In
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
H: ▸ Documents ▸ MATLAB ▸
Current Folder
Name
Command Window
New to MATLAB? See resources for Getting St
>>
Workspace
Name
Value
Class
MATLAB Toolstrip


HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ lavoro ▸ Documents ▸ MATLAB
Current Folder
Name
Command Window
>> 10+90
ans =
100
fx >>
Command History
%-- 30/08...
10+90
Details
Workspace
Name
Value
ans
100
RESULT
LAST COMMAND
VARIABLE IN THE WORKSPACE

1. Familiarizing yourself with the MATLAB desktop

To define a new variable in order to preserve the contents, we can use an assignment statement. For example, create two variables named FIRST and SECOND by typing the following statement in the command line:

```
>> FIRST = 10
FIRST =
         10
>> SECOND = 90
SECOND =
         90
>>
```

```
>> THIRD = FIRST + SECOND
THIRD =
     100
>>
```

1. Familiarizing yourself with the MATLAB desktop

To create an array with ten elements in a single row (row vector), separate the elements with either a space as shown in the following code or a comma (,):

```
>> vector = [10 20 30 40 50 60 70 80 90 100]
vector =
       10   20   30   40   50   60   70   80   90   100
>>
```

Similarly, to create a matrix that has multiple rows, separate the rows withsemicolons, as follows:

```
>> matrix = [10 20 30; 40 50 60 ;70 80 90]
matrix =
       10   20   30
       40   50   60
       70   80   90
>>
```

1. Familiarizing yourself with the MATLAB desktop

To access the elements of an array, use indexing;

```
>> matrix = [10 20 30; 40 50 60 ;70 80 90]
matrix =
       10   20   30
       40   50   60
       70   80   90
>>
```

```
>> matrix (1,2)
ans =
   20
>>
```

To select multiple elements of an array, use the colon operator, specifying a interval of the form start:end.

```
>> matrix (1:3,3)
ans =
   30
   60
   90
>>
```

1. Familiarizing yourself with the MATLAB desktop

Omitting start or end values, we will specify all the elements in that dimension.

```
>> matrix (:,3)
ans =
   30
   60
   90
>>
```

```
>> matrix = [10 20 30; 40 50 60 ;70 80 90]
matrix =
       10   20   30
       40   50   60
       70   80   90
>>
```

The colon operator can also be used to create an equally spaced vector of values using the more general form start:step:end.

```
>> vector_even = 0:2:20
 vector_even =
              0   2   4   6   8   10   12   14   16   18   20
>>
```

1. Familiarizing yourself with the MATLAB desktop

To manipulate the data in the workspace, the following three commands are particularly useful: who, whos, and clear.

```
>> who
Your variables are:
FIRST        SECOND       THIRD        matrix       vector       vector_even
>> whos
  Name             Size             Bytes  Class     Attributes
  FIRST            1x1                  8  double
  SECOND           1x1                  8  double
  THIRD            1x1                  8  double
  matrix           3x3                 72  double
  vector           1x10                80  double
  vector_even      1x11                88  double
>>
```

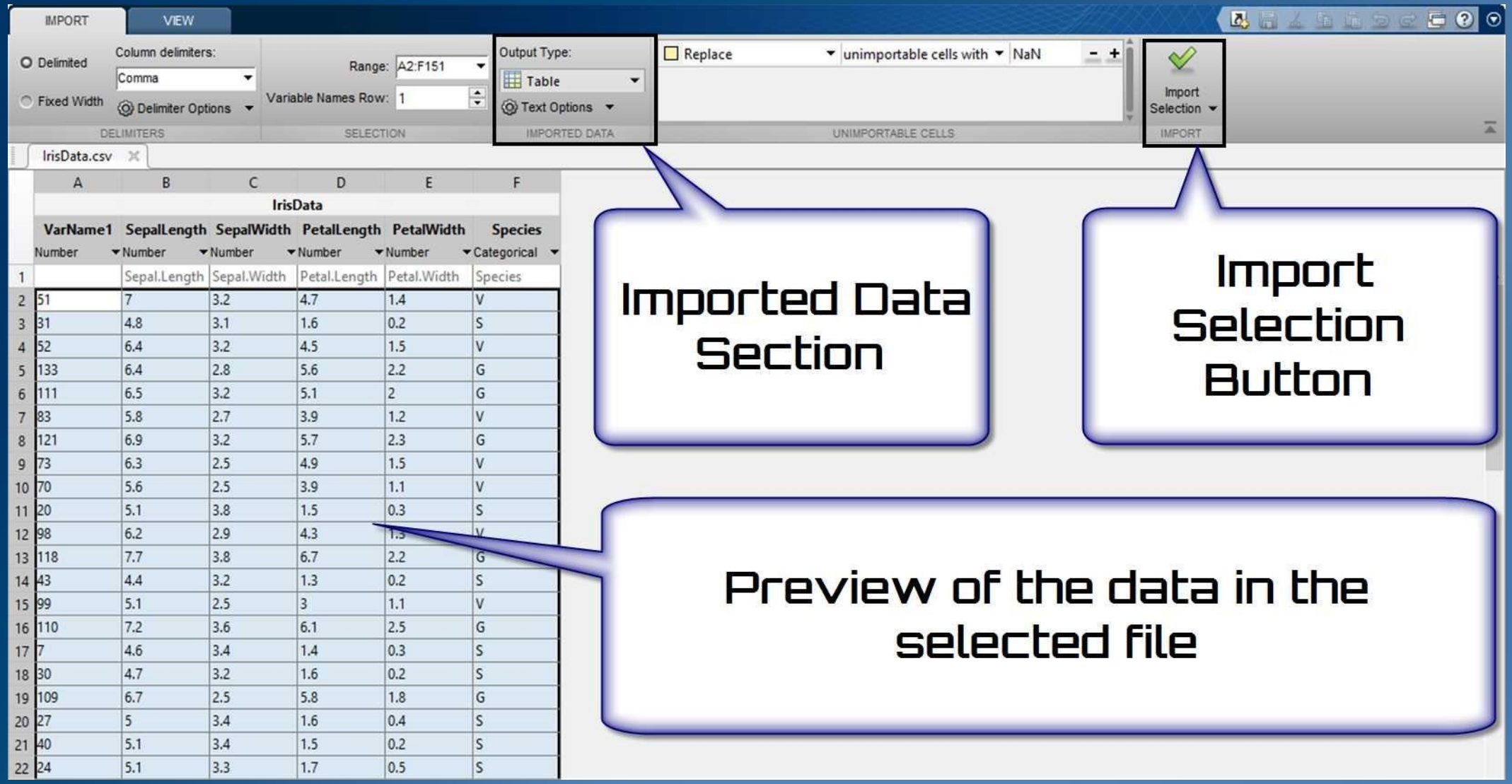
It’s time to tidy up the workspace, removing all of its contents. To do this, we will use the clear command.

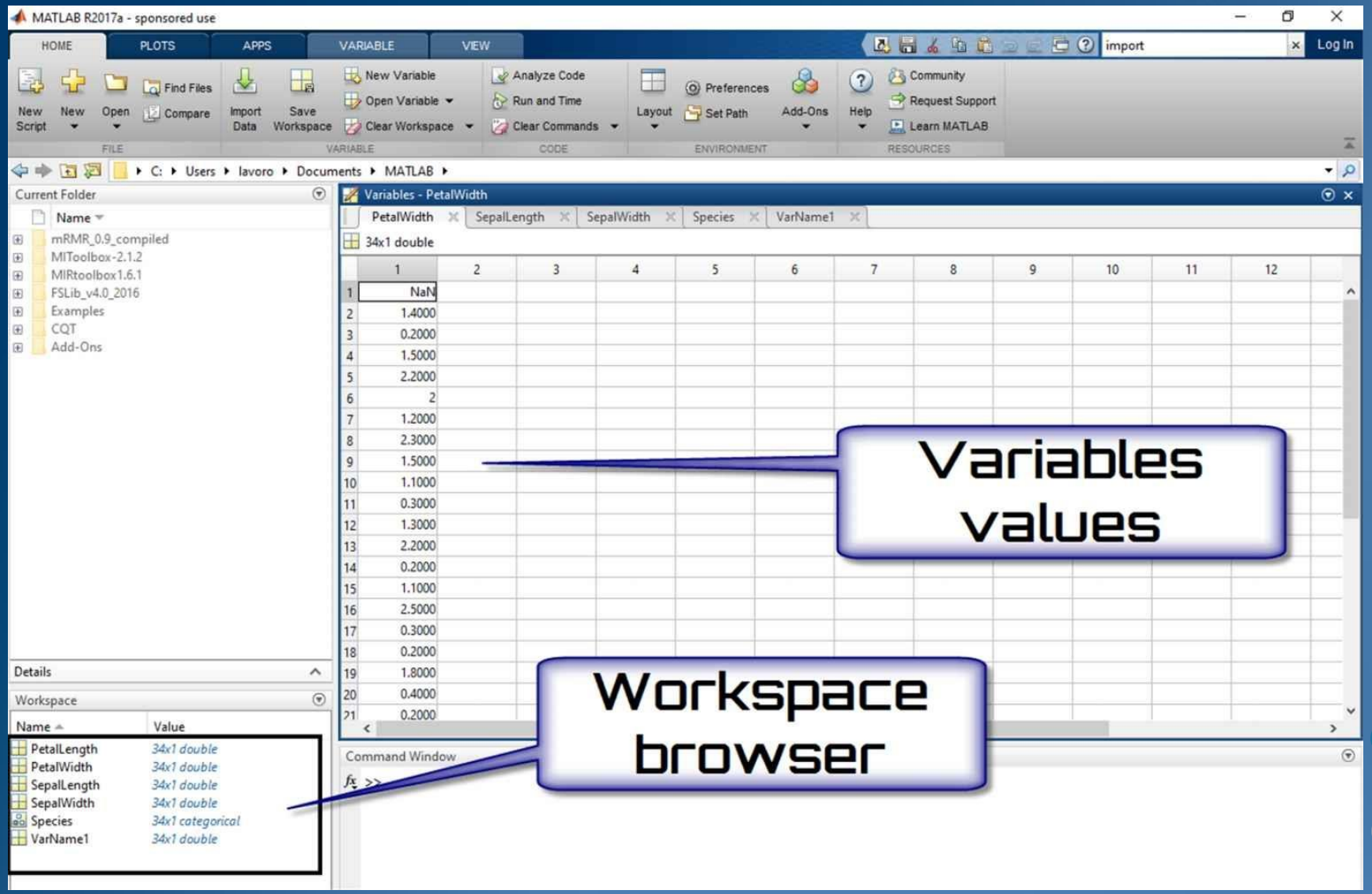
```
>> save filename.mat
```

```
>> load filename.mat
```

```
>> vector
vector =
    10    20    30    40    50    60    70    80    90   100
>> mean(vector)
ans =
    55
>>
```

2. Importing data into MATLAB




MATLAB R2017a - sponsored use
HOME
PLOTS
APPS
VARIABLE
VIEW
import
Log In
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Layout
Preferences
Set Path
Add-Ons
Help
Community
Request Support
Learn MATLAB
FILE
VARIABLE
CODE
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ lavoro ▸ Documents ▸ MATLAB ▸
Current Folder
Name
mRMR_0.9_compiled
MIToolbox-2.1.2
MIRtoolbox1.6.1
FSLib_v4.0_2016
Examples
CQT
Add-Ons
Variables - PetalWidth
PetalWidth
SepalLength
SepalWidth
Species
VarName1
34x1 double
1 2 3 4 5 6 7 8 9 10 11 12
1 NaN
2 1.4000
3 0.2000
4 1.5000
5 2.2000
6 2
7 1.2000
8 2.3000
9 1.5000
10 1.1000
11 0.3000
12 1.3000
13 2.2000
14 0.2000
15 1.1000
16 2.5000
17 0.3000
18 0.2000
19 1.8000
20 0.4000
21 0.2000
Variables values
Details
Workspace
Name
Value
PetalLength 34x1 double
PetalWidth 34x1 double
SepalLength 34x1 double
SepalWidth 34x1 double
Species 34x1 categorical
VarName1 34x1 double
Workspace browser
Command Window
>>

2. Importing data into MATLAB

Importing data programmatically

As anticipated, we can import files into MATLAB programmatically. This way of importing data is very important as it can be inserted into a script and rendered automatically, while the wizard requires the presence of an operator.

2. Importing data into MATLAB

Loading variables from file

```
>> save filename.mat
>> load filename.mat
```

```
>> load matrix.txt
>> matrix
matrix =
    10    20    30
    40    50    60
    70    80    90
>>
```

2. Importing data into MATLAB

Reading an ASCII-delimited file

```
10;20;30
40;50;60
70;80;90
```

```
>> MatrixTxt=dlmread('matrix.txt',';')
MatrixTxt =
    10    20    30
    40    50    60
    70    80    90
>>
```

```
>> MatrixTxt2=dlmread('matrix.txt',',','A1..B2')
MatrixTxt2 =
    10    20
    40    50
>>
```

2. Importing data into MATLAB

Comma-separated value files

```
>> MatrixCsv=csvread('matrix.csv')
MatrixCsv =
    10    20    30
    40    50    60
    70    80    90
>>
```

```
10,20,30
40,50,60
70,80,90
```

```
>> MatrixCsv2=csvread('matrix.csv',0,0,[0,0,1,2])
MatrixCsv2 =
    10    20    30
    40    50    60
>>
```

2. Importing data into MATLAB

Importing spreadsheets

Day	T Mean	T max	T min
1	26	24	29
2	26	24	29
3	26	24	30
4	27	24	30
5	26	23	28

```
>>values = xlsread('capri.xlsx','Temp')
values =
1       26       24       29
2       26       24       29
3       26       24       30
4       27       24       30
5       26       23       28
>>
```

2. Importing data into MATLAB

Importing spreadsheets

```
>>[values,headertxt ]= xlsread('capri.xlsx','Temp')
values =
1     26      24      29
2     26      24      29
3     26      24      30

4     27      24      30
5     26      23      28
>> headertxt=
'Day' 'TMean' 'Tmin' 'Tmax'
>>
```

Day	T Mean	T max	T min
1	26	24	29
2	26	24	29
3	26	24	30
4	27	24	30
5	26	23	28

2. Importing data into MATLAB

Importing spreadsheets

```
>>row1_2 = xlsread('capri.xlsx','Temp',’A2:D3’)
row1_2 =
1      26      24      29
2      26      24      29
>>
```

```
>>column_C = xlsread('capri.xlsx','Temp',’C2:C6’)
column_C =
24
24
24
24
23
>>
```

Day	T Mean	T max	T min
1	26	24	29
2	26	24	29
3	26	24	30
4	27	24	30
5	26	23	28

2. Importing data into MATLAB

Reading mixed strings and numbers

	A	B	C	D	E	F
1	N	Museum	City	Visitors2016	Visitors2015	
2	1	Colosseo e Foro Romano	ROMA	6408852	6551046	
3	2	Scavi di Pompei	POMPEI	3283740	2934010	
4	3	Galleria degli Uffizi	FIRENZE	2010631	1971758	
5	4	Galleria dell'Accademia di Firenze	FIRENZE	1461185	1415397	
6	5	Castel Sant'Angelo	ROMA	1234443	1047326	
7	6	Venaria Reale	VENARIA R.	1012033	580786	
8	7	Museo Egizio di Torino	TORINO	881463	863535	
9	8	Circuito Museale Boboli …	FIRENZE	852095	772934	
10	9	Reggia di Caserta	CASERTA	683070	497197	
11	10	Galleria Borghese	ROMA	527937	506442	
12						

	A	B	C	D	E	F
1	N	Museum	City	Visitors2016	Visitors2015	
2	1	Colosseo e Foro Romano	ROMA	6408852	6551046	
3	2	Scavi di Pompei	POMPEI	3283740	2934010	
4	3	Galleria degli Uffizi	FIRENZE	2010631	1971758	
5	4	Galleria dell'Accademia di Firenze	FIRENZE	1461185	1415397	
6	5	Castel Sant'Angelo	ROMA	1234443	1047326	
7	6	Venaria Reale	VENARIA R.	1012033	580786	
8	7	Museo Egizio di Torino	TORINO	881463	863535	
9	8	Circuito Museale Boboli …	FIRENZE	852095	772934	
10	9	Reggia di Caserta	CASERTA	683070	497197	
11	10	Galleria Borghese	ROMA	527937	506442	
12						

```
>> TableMuseum = readtable('museum.xls')
TableMuseum =
  10×5 table
N    Museum                        City        Visitors_2016 Visitors_2015
__  ____________________________ _________  _____________ _____________

1   'Colosseo e Foro Romano'     'ROMA'      6.4089e+06    6.551e+06
2   'Scavi di Pompei'            'POMPEI'    3.2837e+06    2.934e+06
3   'Galleria degli Uffizi'      'FIRENZE'   2.0106e+06    1.9718e+06
4   'Galleria dell'Accademia…'   'FIRENZE'   1.4612e+06    1.4154e+06
5   'Castel Sant'Angelo'         'ROMA'      1.2344e+06    1.0473e+06
6   'Venaria Reale'              'VENARIA'   1.012e+06     5.8079e+05
7   'Museo Egizio di Torino'     'TORINO'    8.8146e+05    8.6354e+05
8   'Circuito Museale Boboli …'  'FIRENZE'   8.521e+05     7.7293e+05
9   'Reggia di Caserta'          'CASERTA'   6.8307e+05    4.972e+05
10  'Galleria Borghese'          'ROMA'      5.2794e+05    5.0644e+05
```

```
>> TableMuseum = readtable('museum.xls','ReadVariableNames',false)
TableMuseum =
  10×5 table
Var1 Var2                         Var3        Var4          Var5
__  ____________________________ _________  _____________ _____________

1   'Colosseo e Foro Romano'     'ROMA'      6.4089e+06    6.551e+06
2   'Scavi di Pompei'            'POMPEI'    3.2837e+06    2.934e+06
3   'Galleria degli Uffizi'      'FIRENZE'   2.0106e+06    1.9718e+06
4   'Galleria dell'Accademia…'   'FIRENZE'   1.4612e+06    1.4154e+06
5   'Castel Sant'Angelo'         'ROMA'      1.2344e+06    1.0473e+06
6   'Venaria Reale'              'VENARIA'   1.012e+06     5.8079e+05
7   'Museo Egizio di Torino'     'TORINO'    8.8146e+05    8.6354e+05
8   'Circuito Museale Boboli …'  'FIRENZE'   8.521e+05     7.7293e+05
9   'Reggia di Caserta'          'CASERTA'   6.8307e+05    4.972e+05
10  'Galleria Borghese'          'ROMA'      5.2794e+05    5.0644e+05
```

3. Exporting data from MATLAB

```
>> MyMatrix = rand(5)
MyMatrix =
    0.7577    0.7060    0.8235    0.4387    0.4898
    0.7431    0.0318    0.6948    0.3816    0.4456
    0.3922    0.2769    0.3171    0.7655    0.6463
    0.6555    0.0462    0.9502    0.7952    0.7094
    0.1712    0.0971    0.0344    0.1869    0.7547
```

```
>> dlmwrite('MyMatrix.txt', MyMatrix)
```

```
>> type('MyMatrix.txt')
0.75774,0.70605,0.82346,0.43874,0.48976
0.74313,0.031833,0.69483,0.38156,0.44559
0.39223,0.27692,0.3171,0.76552,0.64631
0.65548,0.046171,0.95022,0.7952,0.70936
0.17119,0.097132,0.034446,0.18687,0.75469
```

3. Exporting data from MATLAB

```
>> MyMatrix = rand(5)
MyMatrix =
    0.2760    0.4984    0.7513    0.9593    0.8407
    0.6797    0.9597    0.2551    0.5472    0.2543
    0.6551    0.3404    0.5060    0.1386    0.8143
    0.1626    0.5853    0.6991    0.1493    0.2435
    0.1190    0.2238    0.8909    0.2575    0.9293
```

```
>> xlswrite('MyMatrix.xls', MyMatrix)
```

```
>> MyMatrix = rand(5)
MyMatrix =
    0.3500    0.3517    0.2858    0.0759    0.1299
    0.1966    0.8308    0.7572    0.0540    0.5688
    0.2511    0.5853    0.7537    0.5308    0.4694
    0.6160    0.5497    0.3804    0.7792    0.0119
    0.4733    0.9172    0.5678    0.9340    0.3371
```

```
>> csvwrite('MyMatrix.csv', MyMatrix)
```

4. Working with media files

Handling images

MATLAB provides several functions to operate on and display images. The following list will give a description of the most used functions for image handling:

- `imread`: Read the image from the graphics file
- `imwrite`: Write the image to the graphics file
- `image`: Display the image (create image object)
- `mfinfo`: Get image information from the graphics file
- `imagesc`: Scale data and display as an image
- `ind2rgb`: Convert an indexed image to an RGB image

```
>> Coliseum = imread('coliseum.jpg');
```

```
>>imwrite(Coliseum,'coliseum.jpg');
```

4. Working with media files

Sound import/export

- `audioread`: Read audio file
- `audioinfo`: Information about the audio file
- `audiowrite`: Write audio file
- `audiodevinfo`: Information about the audio device
- `audioplayer`: Create an object for playing audio
- `audiorecorder`: Create an object for recording audio
- `sound`: Convert a matrix of signal data to sound
- `soundsc`: Scale the data and play as sound
- `beep`: Produce an operating system beep sound

5. Data organization

Cell array

A cell array is a datatype that has indexed data containers called cells. Each cell can contain any type of data; cell arrays can contain, for example, text strings, combinations of text and numbers, or numeric arrays of different sizes.

```
>> MyFamily = {'Luigi', 'Simone', 'Tiziana'; 13, 11, 43}
MyFamily =
  2×3 cell array
    'Luigi'    'Simone'    'Tiziana'
    [   13]    [    11]    [     43]
```

```
>> MyFamily2= MyFamily(1:2,1:2)
MyFamily2 =
  2×2 cell array
    'Luigi'    'Simone'
    [   13]    [    11]
```

```
>> LastCell= MyFamily{2,3}

LastCell =
    43
>> class(LastCell)
ans =
double
```

```
>> MyFamily2{2,2}=110
MyFamily2 =
  2×2 cell array
    'Luigi'    'Simone'
    [   13]    [   110]
```

```
>> MyFamily{1:2,1:3}
ans =
    'Luigi'
ans =
    13
ans =
    'Simone'
ans =
    11
ans =
    'Tiziana'
ans =
   430
```

```
>> [r1c1, r2c1, r1c2, r2c2, r1c3, r2c3]= MyFamily{1:2,1:3}
r1c1 =
    'Luigi'
r2c1 =
    13
r1c2 =
    'Simone'
r2c2 =
    11
r1c3 =
    'Tiziana'
r2c3 =
   430
```

```
>> class(r1c1)

ans =
    'char'

>> class(r2c1)
ans =
    'double'
```

```
>> Age = [MyFamily{2,:}]
Age =
    13    11   43
```

```
>> MyFamily=[MyFamily;{'M','M','F'}]
MyFamily =
  3×3 cell array

    'Luigi'    'Simone'    'Tiziana'
    [   13]    [    11]    [     43]
    'M'        'M'         'F'
```

5. Data organization

Structure array

A cell array is a datatype that has indexed data containers called cells. Each cell can contain any type of data; cell arrays can contain, for example, text strings, combinations of text and numbers, or numeric arrays of different sizes.

```
>> MyFamily = {'Luigi', 'Simone', 'Tiziana'; 13, 11, 43}
MyFamily =
  2×3 cell array
    'Luigi'    'Simone'    'Tiziana'
    [   13]    [    11]    [     43]
```

```
>> MyFamily2= MyFamily(1:2,1:2)
MyFamily2 =
  2×2 cell array
    'Luigi'    'Simone'
    [   13]    [    11]
```