Programming with MATLAB

Loading and reading files

Dimitris Voudouris, PhD
d.s.voudouris@gmail.com

Load file with numbers only

To load a *.mat file, you can simply use the command *load*

```
load demodata.mat                 % this will load the file, as long as the file is
                                  in your working directory
```

You can also load text files that contain only numerical data

```
load demodata.txt
```

But text files that also contain text data won’t work with *load*:

```
load demodata_str.txt             % will crash because load does not work
                                  with strings
```

Setting up the path

You can add your datafile to the search path in two major ways

```
addpath(‘XXXX’)        % XXXX should be the complete path of your
                       desired file (e.g., C:\User\MatlabSeminar\demodata.mat)

load XXXX              % again, XXXX should be the complete path of your
                       desired file (e.g., C:\User\MatlabSeminar\demodata.mat)
```

Load files with text

If you want to load a file that contains strings, then *load* won't work

Try:

```
load demodata_str.txt                    % will crash
```

Some convenient ways to load such files include:

readmatrix

readtable

Load files with text

Using the function *readmatrix*

```
data = readmatrix(‘demodata_str.txt’)
% no additional inputs required as long as the file is in your path
% MATLAB will just get rid of any text
```

You can access the elements of the matrix *data* as usual:

```
somevariable = data(:, 3);
my_averaged_variable = mean(somevariable)
```

Load files with text

Using the function *readtable*

```
data = readtable(‘demodata_str.txt’)
% no additional inputs required
```

If you have strings in row 1, these will be presented as headers

```
data(:, 3)              % will include the header information
                        % this means, you cannot do numerical operations…
```

Load files with text

Consider only numerical values when using *readtable*

Option 1 – use curly brackets

```
data{:, 3}              % choose all rows from column 3, excluding header
```

Option 2 – use the dot system

```
data.intensity          % choose all rows from column with header ‘intensity’
```

Option 3 – use header name as column input

```
data(:, ‘intensity’)    % choose all rows from column with header ‘intensity’
```

Load files with text

Consider only numerical values when using *readtable*

Option 1 – use curly brackets

```
data{:, 3}                 % choose all rows from column 3, excluding header
```

Advantage:

- Easily choose many columns by specifying the input after the comma

Disadvantage:

- Prone to errors when choosing column, say, 241 from a huge matrix

Load files with text

Consider only numerical values when using *readtable*

Option 2 – use the dot system

```
data.intensity          % choose all rows from column with header ‘intensity’
```

Advantage:

- Use the exact header name to choose that column; straightforward
- Easy to choose the desired column from a huge table

Disadvantage:

- Bulky to read several columns

Load files with text

Consider only numerical values when using *readtable*

Option 3 – use header name as column input
data(:, ‘intensity’)   % choose all rows from column with header ‘intensity’

Advantage (as the dot system):
- Use the exact header name to choose that column; straightforward
- Easy to choose the desired column from a huge table

Disadvantage (as the dot system):
- Bulky to read several columns

Interim suggestion

To get a better grasp of the commands:

Type *(do not copy/paste)* the commands

Execute commands line-by-line and carefully look at the outcome

Execute commands using different input:

predict output before seeing it. Do you and MATLAB agree?

Make the commands fail:

use different brackets, commas, impossible inputs, etc.

The end

Read your own file and apply operations as we learned

Suggestions for open-data sources:
ourworldindata.org
data.europa.eu
open.nasa.gov/open-data
archive.ics.uci.edu
sleepdata.org
kaggle.com/datasets