

Ungraded Matlab programming practice – I recommend you review this material by January 30

Let us first define a matrix:

```
Mat1 = [1 3 -2 4 -6 -1 10 8; 3 5 -6 -2 1 4 0 6; 5 3 1 -1 2 0 2 -2];
```

This matrix has 3 rows and 8 columns.

Question 1:

Create a new matrix `Mat2` that contains the even-numbered columns of all three rows of `Mat1`. In other words, replace `???` with the proper syntax below:

```
Mat2 = Mat1(???)
```

to result in the `Mat2` with the content:

```
[3 4 -1 8; 5 -2 4 6; 3 -1 0 -2]
```

Note the Matlab syntax for a `for` loop is:

```
for i = start:finish
    statement1;
    statement2;
    ...
    statementn;
end;
```

Where the variable `i` increments from the start value to the finish value as the `for` loop repeats.

Question 2:

Create a new matrix `MatDiff` – 1 row, 8 columns – containing the column-wise differences between the elements of row 1 and row 2 of `Mat1`. The contents of `MatDiff` will be:

```
[-2 -2 4 6 -7 -5 10 2] --- 1st element: 1-3 = -2, 2nd element: 3-5 = -2, etc.
```

Question 3:

Create a new matrix `MatDiffSquare` – 1 row, 8 columns – containing the square of each element in `MatDiff`. The contents of `MatDiffSquare` will be:

```
[4 4 16 36 49 25 100 4]
```

Note exponentiation in Matlab is represented by the `^` operation. $4^2 = 16$ and $10^3 = 1000$.

Question 4:

Compute the sum of the squared differences. In other words, find the sum of the elements of `MatDiffSquare`.

Monitoring our Matlab environment:

The syntax for monitoring our working environment in Matlab is discussed in parts of slides 6 and 7 of the Matlab Supplement lecture.

To list the variables already defined in our current Matlab session, type

```
who
```

into the Matlab prompt. This will list the names of the existing variables. You can delete all Matlab variables from the current session using the command

```
clear
```

To determine what computer directory Matlab currently is operating in, type

```
pwd
```

You can see any files in your current operating directory. You can change your operating directory either through the graphical user interface (if you have one) or through the command

```
cd
```

For more guidance on directory structure, as framed in the Linux environment, see <https://cvw.cac.cornell.edu/Linux/files>

Writing functions in Matlab:

The syntax for writing functions in Matlab is discussed in slide 8 of the Matlab Supplement lecture. To create a function named `newFunc`, we save a file named `newFunc.m` in the current working directory of Matlab. Let us say the purpose of `newFunc` is to find the cubed value of a single numeric input. The contents of this function will appear as follows

```
function out = newFunc(in)
    out=in^3;
```

The file begins with the line:

```
function outputName = functionName(inputName)
```

This establishes that the following code is part of the function with the name `functionName`, and that this function takes in the inputs named in `inputName` and that this function outputs the value(s) named in `outputName`.

Over the course of the function, the value(s) to be output is calculated based on the statements in the function. In Matlab, the values named in the first line of the function are automatically returned at the termination of the function, without requiring an official `return` statement.

We **call** the function `newFunc` in Matlab using standard programming syntax, e.g.:

```
x = newFunc(4)
```

will return the value 64 into the variable `x`.

You can download `newFunc.m` from our course website and try it out. Alternatively, on the Fordham erdos computer system, you can get a copy of `newFunc.m` by typing

```
cp ~dleeds/MLpublic/newFunc.m .
```

at the Linux prompt (note, you must include the final `.`).

Once you have tried out `newFunc`, proceed to write the following functions for yourself:

Question 2F: Write function `computeDiff` to compute the difference between the top two rows of the input matrix, as in Question 2.

Question 3F: Write function `computeSquare` to compute the square of each element of a $1 \times N$ matrix, as in Question 3.

Question 4F: Write function `sumElements` to compute the sum of the elements of a $1 \times N$ matrix, as in Question 4.

Loading and saving data in Matlab:

The syntax for loading and saving data in Matlab is discussed in slide 7 of the Matlab Supplement lecture. To load a file from the current working directory in Matlab, type

```
load filename.mat
```

Once the data has been loaded, you can observe the new variables in the Matlab session with the command `who`.

Download `sampleData.mat` from our course web site. Alternatively, on the Fordham erdos computer system, you can get a copy of `sampleData.mat` by typing

```
cp ~dleeds/MLpublic/sampleData.mat .
```

at the Linux prompt (note, you must include the final `.`).

Question 5: Load `sampleData.mat` into Matlab. What new variable is available in your session?

Question 6:

- Compute the “sum-squared difference” between the first two rows of your newly loaded matrix.
- Compute the “sum-squared difference” between the rows three and four of your newly loaded matrix.