# Lecture 1: Matlab’s IDE and Matrix Slicing and Indexing

- Understand the Matlab IDE. Command Window, Editor, Workspace, Variable Editor, Ribbon.
    - 15-minute exercise: Plot an array as a bar graph!

- Arrays/Matrices are the default variable type. To understand Matlab, you must be comfortable with handling arrays and matrices.
    - Even a single number, to Matlab, is a 1x1 array.
    - To make an array, put a list of numbers in square brackets [ ]. ex: [1 3 5 2 6]. You can also specifiy a sequence of numbers with the colon.
        - ex: [1:6] makes the list: [1 2 3 4 5 6].
        - ex: [1:2:10] makes a list that goes up by twos, returning: [1 3 5 7 9].
    - To make a matrix with rows and columns, make a list of numbers that will appear in the same row, then start the new row using the semicolon.
        - ex: [1 2 3; 4 5 6] returns a 2 x 3 matrix.
    - Every matrix has a map associated with it that refers to **where** each number is located in its matrix. Each number’s location is called its **index**.
    - You can “**Slice”**, or **“Index”**, parts of the matrix out and perform operations on them. This is done using the parentheses () after the matrix name. Ex: MyMatrix(1:5)
        - Order: Rows, Columns, 3rd dimension, etc. In this class, we will only use two-dimensional matrices, because they are the most common and are *way* easier to use.
            - All values also have a single-value index, as they can be referred to as a really long array. ex) Try MyMat(6) on a 4x4 matrix.
        - **The Colon Operator ‘:’**:
            - In Matlab: the colon can mean “**through**” (as in ‘1 *through* 5’, when making an array with a sequence). The colon can also mean “**all**”, when alone.
            - When indexing matrices, you *must* use the colon to explicitly tell matlab you want the entire column, or the entire row. You cannot index a row without also indexing a column.

        - 15-minute Drill: What should you enter to slice out different parts of a matrix?

    - **Indexing with Variable-Set Arrays**: Because arrays can be saved to variables, the variables themselves can be used for indexing by substituting the variable name for the array.
        - This is often done, as it allows multiple data sets to map onto each other.
        - ex) if **IndexList** = [1, 3, 6, 7], and **data** = [30:0.5:40], you can slice out

the elements of 'data' listed in 'IndexList' by entering:

        - data(IndexList)

- **Logical Indexing** is a really convenient way to slice matrices in ways that aren't contiguous. In logical indexing, you only slice out parts of a matrix that are **true** (Matlab will show this as 1. False is 0) for a given logical statement.
    - Logical statements use >, <, ==, >=, <=, &, or |.There are 2 oft-used ways to do logical indexing/slicing in Matlab:
        - Logical Statement inside the parentheses: To slice out parts of a matrix that are true, just put the logical statement inside the parentheses.
            - This will return an array of numbers containing the values that are true.
            - For complex statements, though, this can get a bit messy.
    - 15-minute Drill: Using the same matrix as before, find the following values: MyMat >4, <10, >4 && <12, ==6, <= 7, etc

Arrays and Variables (cont'd)

- Matlab is designed to work for matrix algrebra directly, which makes advanced calculation very simple. However, it means that the defaults operators are a bit unintuitive, as they default to matrix algebra concepts.
    - Important example: ***** is Matrix Multiplication, and **.*** (a period, then an asterisk) is Element-Wise Multiplication**.**
        - This distinction is important when wanting to divide (/), multiply (*), or power (^) a matrix or array with another matrix or array. As a general rule, place a period before each of them every time use these operators, unless you expressly desire to do matrix algebra.

- Name your variables carefully:
    - Capitalization matters.
        - ex) "data" and "Data" are two different variable names.
    - Numbering matters. Numbers can be used in the variable name, but must not be the first character.
        - ex) "data1" and "data2" are different variables,
        - ex) "1data" is an illegal name, but "d1ata", "dat1a" and "data1" are not.
    - The Name Matters. Remember that when you name a variable, you overwrite any other variables or even functions that share the same name.
        - Tip: Before naming a variable, type 'help variableName'. Not only will you see if the variable already exists, but you might learn how to use a new function as a bonus!

Name: ________________________________

Take-home worksheet: Due at Start of Lecture 2

The following worksheet is meant to help you get comfortable with matrix slicing and indexing in Matlab. Please fill in the blanks for the following questions by writing the actual Matlab code, and have Matlab open in front of you while answering them to check your answers. If you are not comfortable with this part of Matlab by the end of the worksheet, please go through the lecture notes and re-do the in-class exercises in Matlab until you are comfortable with them.

Make a matrix **MyMat = magic(6);.** What do you type to get:

1. The third row of MyMat? ________________________
2. the second column of MyMat? ________________________
3. The 8th number in MyMat? ________________________
4. The 4th and 6th row of MyMat? ______________________
5. The 4th through 6th column of MyMat? ____________________
6. Every 3rd column of MyMat? ____________________
7. All values of MyMat greater than 11? ________________________
8. All values of MyMat equal to 7? ________________________
9. All values of MyMat less than the third value of MyMat? _____________________

An example experiment has 30 subjects, men and women. You record their average reaction time (in seconds), looking for differences between the two genders.

- Let's make some data to analyze from this example experiment:
    a. **participants** is the list of participant numbers. To make it, type **participants = [1:30]'**
    b. **gender** is the list of genders of each participant. To make it, type **genderList = randi(2,[30 1])**
    c. **reactionTime** is the data to analyze, containing a list of each participant's average reaction time. To make it, type **reactionTime = rand([30 1]]**

1. Answer the following questions about this experiment: What do you type to get:

    a. The reaction time of the 11th participant? ______________________________

    b. The gender of the 5th participant? _______________________________

    c. All participants that were women? ______________________________

    d. The reaction time of all men participants? ____________________________

    e. The participant numbers of all people with reaction times longer than 0.5 meters? __________________________

2. What would you type to make one matrix out of the three arrays **participants**, **conditionList**, and **reactionTime**, which has 30 rows (one for each participant) and three columns (the first column with participant numbers, the second column with their condition number, and the third column with their reaction time)?

    ___________________________________________________________