PSYC40012 Models of Psychological Processes

Tutorial Video 1

In this video, I am going to review some basic Matlab commands so that you can follow along with the use of Matlab in the MPP lab classes.

Matlab is a high-level programming language, which means that it is easier to test your code and program without first compiling your code (as in other programming langauges such as C++ or Fortran). This makes programming more intuitive, and once you've acquired some familiarity with the basics, you'll be able to understand and complete the programming tasks necessary for the course.

Matlab is a commercial software package which means that you need to pay to use it. The University of Melbourne has a license which we will use in class. If you want to get Matlab at home, there are student versions available which are restricted versions of the full version (but that won't matter for this class). There is also another programming language called Octave which is open source and free but uses the same syntax and commands as Matlab.

Now I will go through commands and show you how to do basic programming in Matlab. (There will be a "cheat-sheet" of this tutorial available on the course LMS site which you can download and follow along with or review afterward).

---

**Elementary operations**

```
+, -, *, /, ^
```

---

**Relational operations**

```
<, >, <=, >=

==, ~=
```

---

**Logical operation**

```
&&    % AND

||    % OR

xor(1,0)
```

You may be unfamiliar with logical operations so I have added a link to the following chapter on LMS. The chapter is from a book which describes in very general terms how computers work, and the chapter on logic gates is an excellent introduction. I recommend the entire book

Petzold, C. (2000). Chapter 10: Logic and switches. *Code: The Hidden Language of Computer Hardware and Software. (p. 86-101).* Redmond, WA: Microsoft Press.

---

The AND and OR functions are reasonably intuitive to understand (both inputs have to be true for the AND function to be true; only one of the inputs needs to be true for the OR function to be true). The XOR function is a little less intuitive; however, as we shall see in the coming weeks, the XOR function is a very important function in psychology. In order to give you an intuitive understanding of this function, let me give you an example of where you might have experienced an XOR function in your own life:

*Imagine that you live in a house with a large dining room, which you can enter from either the living room on the north side or the kitchen on the south side. There are two light switches located at either entrance. It is impractical to have only a single switch as you would have to cross the room each time you wanted to turn on the light so both switches control the main light in the room.*

*If the light is currently off and you enter from the living room, flicking the living room switch will turn the light on. If you walk through the dining room and hit the switch at the kitchen entrance the light will go off again. You can perform the same action in the other direction, starting in the kitchen and ending in the living room. Both switches were initially up and the light was off. At the end of your progression through the dining room, both switches are now down. In both cases, the light is off. So if both switches are up or both switches are down, the light is off. If one switch is up and the other is down, the light will be on. This is exactly what an XOR operation does.*

*You'll note that the lights at the top and the bottom of the stairs between each floor in the west end of the Redmond Barry building operate in this manner. Try them out sometime!*

**Creating Variables**

```
x = 3
```

You can use a semi-colon at the end of the statement to suppress output to the screen

```
y = 'Psychology' % Create strings using single quotes
z = (6 < 3) % Create logical variables
a = pi  % Store variables in the workspace
a       % prints a on its own
```

---

**Creating matrices**

```
X = [7 6; 5 4; 3 2]
```

The semicolon means “start the next row of the matrix”

```
X = [7 6;
5 4;
3 2]
```

We can also create row vectors (which have 1 row and a variable number of columns)

Row vector (1 x 3)

```
r = [4 5 6]
```

Column vectors have a variable number of rows and a single column

Column vector (3 x 1)

```
c = [1; 2; 3]
```

We can also use the transpose operator to change a row to a column and vice versa

```
c = [4 5 6]’
```

We can create vectors using the colon operator

```
v = 1:0.5:10
```

This creates a vector, called v, with a bunch of elements that starts at 1 and goes to 10 in steps of .5

```
v = 1:6
```

If you don't specify a step then the step size is 1

---

Matlab has many built-in functions which we will discuss in more detail in the coming tutorial videos. There are several functions which you can use to create matrices and vectors:

```
ones(3,4)     % creates a 3 x 4 matrix of ones
```

Note how the number of rows comes before the number of columns. This will always be the case in Matlab

```
C = 7*ones(3,4)

C = [7 7 7 7; 7 7 7 7; 7 7 7 7]
```

---

```
A = ones(1,3)    % row of three ones

B = zeros(1,3)   % row of three zeros

C = rand(1,3)    % Creates a vector of random numbers
```

C is a 1 x 3 matrix of all random numbers from a uniform distribution between 0 and 1. The uniform distribution means all numbers between 0 and 1 are equally likely to occur

```
D = randn(1,3)  % Creates a vector of normally distributed
random nubmers
```

D is a 1 x 3 matrix of normally distributed random numbers with mean 0 and s 1

---

Using Matlab we can create more complicated expressions

```
X = 10 * sqrt(1:3) + randn(1,3)

IQ = 100 + 10 * randn(1, 10000)
```

(IQ has a mean of 100 and a standard deviation of 10)

We can use Matlab to plot our variables

```
hist(IQ)  %  plot IQ with 10 bins (default)

hist(IQ,50) % plot IQ with 50 bins
```

There are many other Matlab plotting functions; I will talk about some of these in upcoming videos.

---

**Getting help**

Notice we've used a number of functions, such as `sqrt, rand,` and so on

If we want to know what a function does, we can type:

```
help sqrt
doc sqrt
```

In the next video we will discuss how to access and use variable in Matlab in a more sophisticated manner.

I would recommend that you don't try to memorize all of these functions; instead, use this document (and the other cheat sheets) as a guide and look up what you do not remember. Also make extensive use of the help menu. It will save you a lot of time if you make the effort to understand the information in the help menu when you need the answer to some specific question.