

Video: “Working with arrays and variables” (6:22 mins)

(0:00)

This lesson discusses the basics of working with data on MATLAB. When scientific data is represented in tabular form and MATLAB is designed to work with tables very easily. We will talk about arrays and how them to represent tables, we will need to go over a few array concepts to start with such array dimension and the correspondence of dimension to rows and columns. Then we will talk how to get an individual element out of an array, how to split an array into its individual rows and columns, introduce the idea of a variable and variable assignment. Finally, we will show how to enter data in MATLAB.

(00:44)

MATLAB uses arrays to represent tabular data. The table on the right shows the number of cases in thousands of measles that occurred in New York City from the years 1931 to 1936. The first column of the table shows the year the second column shows the corresponding number of cases. MATLAB represents the data using an array which is a rectangular arrangement of data values. This array is two dimensional it has rows and columns.

(1:15)

Higher dimensional arrays are also possible. For example, if 3D arrays consist of a stack of 2D arrays we mainly consider 2D arrays, that is arrays of dimension two. MATLAB specifically identifies the rows as dimension one and the columns as dimension two. In order to access array value, we have to associate the array with the name meaning with assign it to a variable, we will talk about how to do that in a minute. Once we assign the array to a variable, we can refer to any item by the variable name and the position of the item in the array. Row and column number start with one. In the example the variable name is disease which holds a 6x2 array the value 1931 has indices 1,1 because it is row 1 column 1. The value 26.5 has indices 1,2 because it is in row 1 column 2.

(2:11)

Now let’s look at how to pick out a particular item from an array. The strategies to give the name of the variable holding an array followed by the row and column number of the item is much the same way as you give an address a street name and then a number. In the example, disease 1, 2 shows the thousands of cases for 1931. Just having the value by itself is not very useful usually we will use it as part of a computation were, we will store the value in a variable for future use. In the second example, we are using cases 1931 to hold the value. We should pick variable names that help you remember what the value represents. The handout for this video gives a few rules about variable names.

(2:58)

You can use the same strategy to pick out entire rows and columns from an array. Instead of putting a specific number use colon to designate all rows or all columns. For example, to pick up column 2 of disease you put 2 in for the column selector. We want all rows in column 2 so we use colon for the row selector. In the second example, we create a new variable “cases” to hold column two of disease.

(3:32)

Often the data we analyze will be stored in files and we use the load command to bring into MATLAB and assign the variables to a variable. This example shows how to directly enter the value- array values are listed from right to left top to bottom the values are enclosed in square brackets. The values in the same row are separated by columns or blanks and semicolons start a new row shown by this example. In the example we did not have much room so the statement when over multiple lines. We used three periods in a row to indicate the statement continues to the next line.

(4:10)

Before demonstrating these ideas in MATLAB, let's talk briefly about the assignment statements used in these examples. A equals B does not mean that A and B are the same. Either equal represents assignment that is a values compute to expression B and assigned to variable A. It can represent a complicated mathematical formula that A should be a variable. In the example expression B on the right of this assignment statement extracts column two from disease. This column of values has been associated with the variable cases.

(4:50)

Now let's demonstrate these concepts in MATLAB, we are going to enter the commands on the command window rather than creating a script. We will begin by creating a variable "MyCases" and assigning it an array to explicit values. I start with a square bracket, the values on the first row are separated by commas when the first-row ends, I enter a semicolon then I begin to enter the second row separated by commas followed by a semicolon. And finally, the last row deprived by commas and ending with a square. End the whole statement with a semicolon to suppress printing and hit enter.

(5:33)

When you hit enter "MyCases" appears in the workspace and if I double click on it, I can see the actual values in variable editor. Indeed, I have an array with three rows and two columns. Now let me create a variable "cases" that is the second column of "MyCases" this time I do not put a semicolon at the end but when I hit enter, I actually see the values displayed in the command window. Again, cases appear in the workspace. Finally, I am going to enter the variable "cases1931" as "MyCases 1, 2" designating the entry on the first row second column.