

Topic for the class-Octave
Unit _4 : Title-Data Science tools
Date & Time : 11.10.24 10.00 AM – 10.50 AM

Dr. Bhramaramba Ravi

Professor

Department of Computer Science and Engineering

GITAM School of Technology (GST)

Visakhapatnam – 530045

Email: bravi@gitam.edu

Unit4-syllabus

UNIT 4

Data Science tools

9 hours, P - 2hours

Overview and Demonstration of Open source tools such as R, Octave, Scilab. Python libraries: SciPy and sci- kitLearn, PyBrain, Pylearn2; Weka.

<https://www.coursera.org/learn/open-source-tools-for-data-science>

Octave

- Octave is a high-level programming language primarily used for numerical computations. It is largely compatible with MATLAB, making it a popular alternative for those who want to work with numerical computing but do not have access to MATLAB.

What is Octave

- **OCTAVE** is a high-level language designed for scientific computations. It is most compatible with MATLAB in speed and syntax (to some extent).
- Octave can be downloaded from <https://www.gnu.org/software/octave/download.html#ms-windows>. It features a simple-to-use GUI for scripting files and a command line for instant execution of a line of code. Since it is based on MATLAB, Octave produces *M-files*, i.e., **.m** extension files.
- It is free open source software under GNU General Public License. It was made for computing linear algebra and matrices. Now, it is used for making machine learning models.

Advantages of using Octave

- There are many other advantages of using octave:
 - 1.It has C, C++ and FORTRAN as its base languages, which makes it a lot faster.
 - 2.Its syntax is very close to mathematical expressions which makes it a good choice for engineers.
 - 3.It has high level plot commands for 2D and 3D, for data visualization.
 - 4.It supports contour plots.

Useful things to know

- So, in Octave, maybe the execution of your results into an infinite iteration then you can type `ctrl + C` to terminate the process. You can terminate any process with this shortcut.
- One-line commenting in Octave either using `#` or `%`.
- `>> % I am a one-line comment`

Useful things to know

- Block commenting in Octave : `{ }` or `%{ ... %}`.
- A line of code is preferred to end with semicolon `;` to stop the instant execution.
- There is one more command that will help you to understand Octave. It is about taking help from Octave by typing *help* or *doc* followed by a command about which you wanted to know. This command opens the documentation of the command in the command window or command-line interface (CLI).

Help command

```
>> help clc                                % CLC is a command in Octave
'clc' is a built-in function from the file libinterp/corefcn/sysdep.cc
  -- clc ()
  -- home ()
```


Clc command

Clear the terminal screen and move the cursor to the upper left corner.

...

Octave

- Octave GUI has a documentation window as well where you can find many functions and commands.
- To close Octave, type *exit* or *quit*.

Key Features of Octave:

- 1.Syntax Compatibility:** Octave's syntax is very similar to MATLAB, allowing scripts to be run on either platform with minimal modification.
- 2.Interactive Environment:** It offers an interactive command-line interface and can also run as a script-based program.
- 3.Numerical Capabilities:** Octave provides functions for linear algebra, data manipulation, optimization, statistics, and numerical integration.
- 4.Visualization:** Supports 2D and 3D plotting using tools like gnuplot or its built-in plotting libraries.
- 5.Open Source:** Octave is free and open-source software, making it accessible for everyone.

Common Use Cases:

- Mathematical modeling and simulations.
- Data analysis and visualization.
- Scientific research and engineering computations.

Variables and data types

- **Variables** are x and y that we have been using since our junior classes. We store values in variables and the type of the value stored is called the Data **type** of the variable.
- Like other programming languages, Octave also provides a simple way to declare variables with the following conditions:
 - 1. It must be a sequence of letters, digits and underscores, not beginning from digits. So, **my_var12**, **my21var_**, **_var12** and **_123var** are valid variables but **21my_var** is an invalid variables.
 - 2. It should be kept in mind that the Octave is case-sensitive. So, **myVar** and **myvar** are treated as different variables.
 - 3. To check the validity of a variable name, there is a function to use: **isvarname** ('variable_name'). This returns Boolean value, i.e., true (**1**) or false (**0**).
 - 4. Octave has an automatic variable: **ans**, which appears on pressing "*Enter*" after writing a line of code.
- Octave has some standard built-in data types for real and complex scalars, matrices, ranges, characters, a data structure type, and cell arrays. For example, "scalar", "matrix", etc. The function *typeinfo (expr)* returns the data type of the *expr* in string form.
- As said earlier, Octave works very fast with vectors and matrices.

Useful terms

- **Matrices** are a collection of entities in some m rows and n columns. **Vectors** or **arrays** are matrices with one column or row. **Scalars** are matrices with dimension 1×1 , i.e., a scalar is a single entity matrix. The **entities** are categorized into **integers**, **characters** and **logical**. **Integers** are set to “double” by default which has the largest range of numbers. Other values for integers are “int8”, “int16”, “int32”, “int64”, “uint8”, “uint16”, “uint32”, “uint64” and “single”. **Characters** are single/double quoted characters like ‘ab’, ‘cdac’, ‘1’, etc. while **logicals** are binary values, i.e., **0** (false) and **1** (true)

Creating a matrix

- To create a matrix, simply type

```
>>A=[5, 4, -3; 9, -6, 7; 2, -3, 4]
```

```
A =
```

```
5 4 -3
```

```
9 -6 7
```

```
2,-3, 4
```

```
>>
```

Create a vector


- To create a vector, we just need to replace semicolon inside square braces with spaces.

```
>>V=[5 4 -3 6 7 -9 9 7 6]
```

```
V = 5 4 -3 6 7 -9 9 7 6
```


Basic Example in Octave:

octave

 Copy code

```
% Define a matrix
```

```
A = [1, 2; 3, 4];
```

```
% Perform a matrix multiplication
```

```
B = A * A;
```

```
% Display the result
```

```
disp(B);
```