# Review of Lecture 10

- **Example 1: BMI calculator**

```
function bmi = BMIcalculator (weight, height)
```

- **Example 2: Population growth**

```
function pop = popGrowth (initialPop, growthRate, time)
```

- **Example 3: Ohm’s law calculation in electrical circuit**

```
function result = ohmsLaw (curr, resist, volt)
```

- **Example 4: Projectile motion calculation**

```
function distance = projectileD (speed, angle)
```

- **Example 5: Factorial of a number**

```
function f = facto(n)
```

- **Example 6: Guessing number Game**

```
function guessingNumber()
```

## Info 3

# Introduction to MATLAB®

**M. Bouzenita**
2nd year Engineer - University of Jijel

## Lecture 11

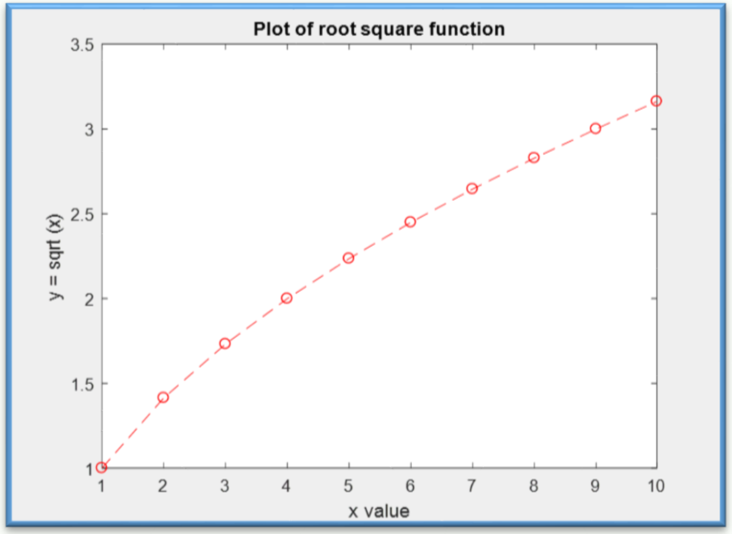
# Graphics in MATLAB(1/2)

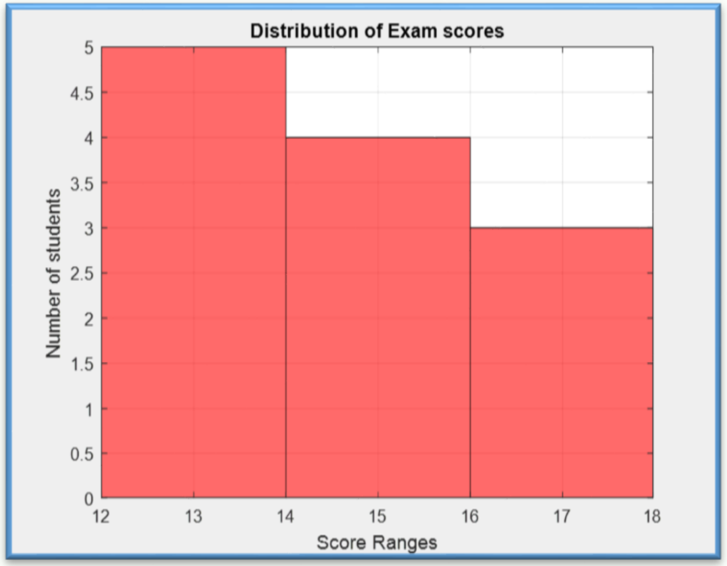
# 1. Introduction

Visualization is one of the important tools to **judge** and **analyze** the numerical data and **provide meaningful information** about the data under study.

MATLAB provides such feature to visualize data in different manners including, 2D plot, 3D plot ...

Furthermore, MATLAB provides additional information with plots such as **title**, **labels of axes**, **labels for data, grids** and other useful information.



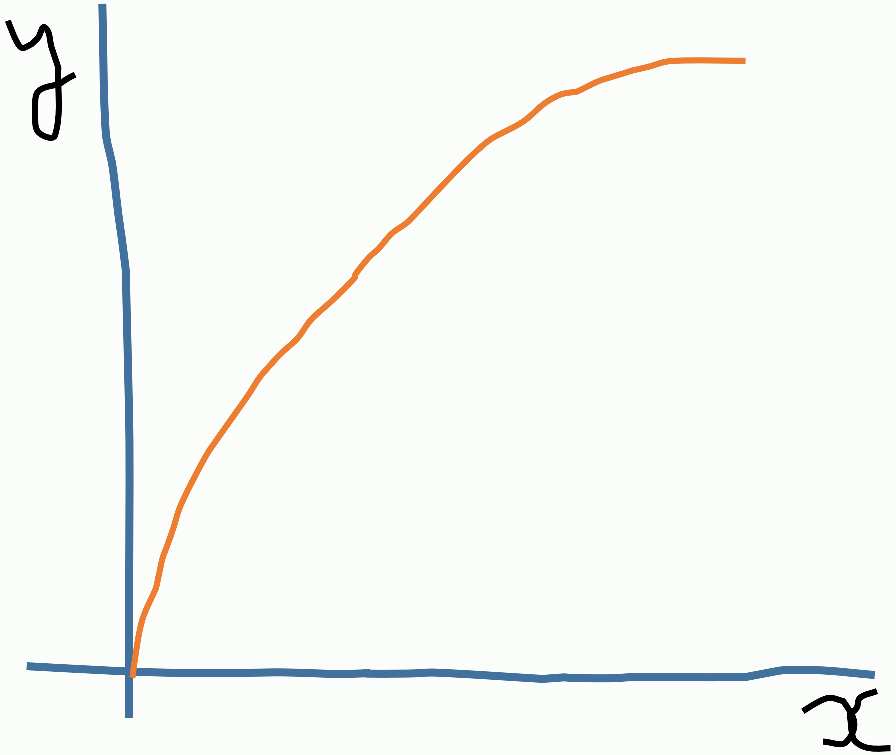


# 2. 2D plot

```
plot (X, Y)
```

Creates a 2-D line plot of the data in Y versus the corresponding values in X.

$$y = f(x)$$



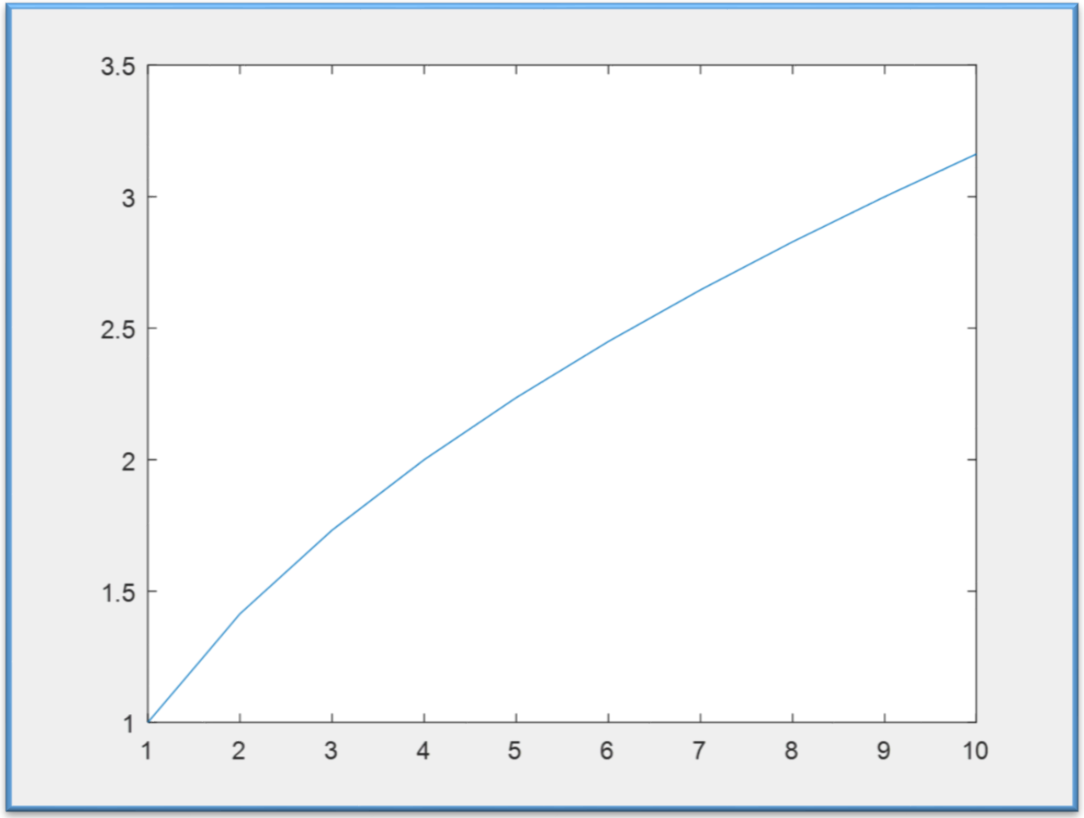
# 2. 2D plot

`plot (X, Y)`

Creates a 2-D line plot of the data in Y versus the corresponding values in X.

The following example shows a script file introduced to plot the **root square** function in the interval [1 10]: $y = \sqrt{x}$

```
>> X = 1:10;

>> Y = sqrt(X);

>> figure;

>> plot(X, Y);
```



# 2. 2D plot

**`plot (X, Y)`**

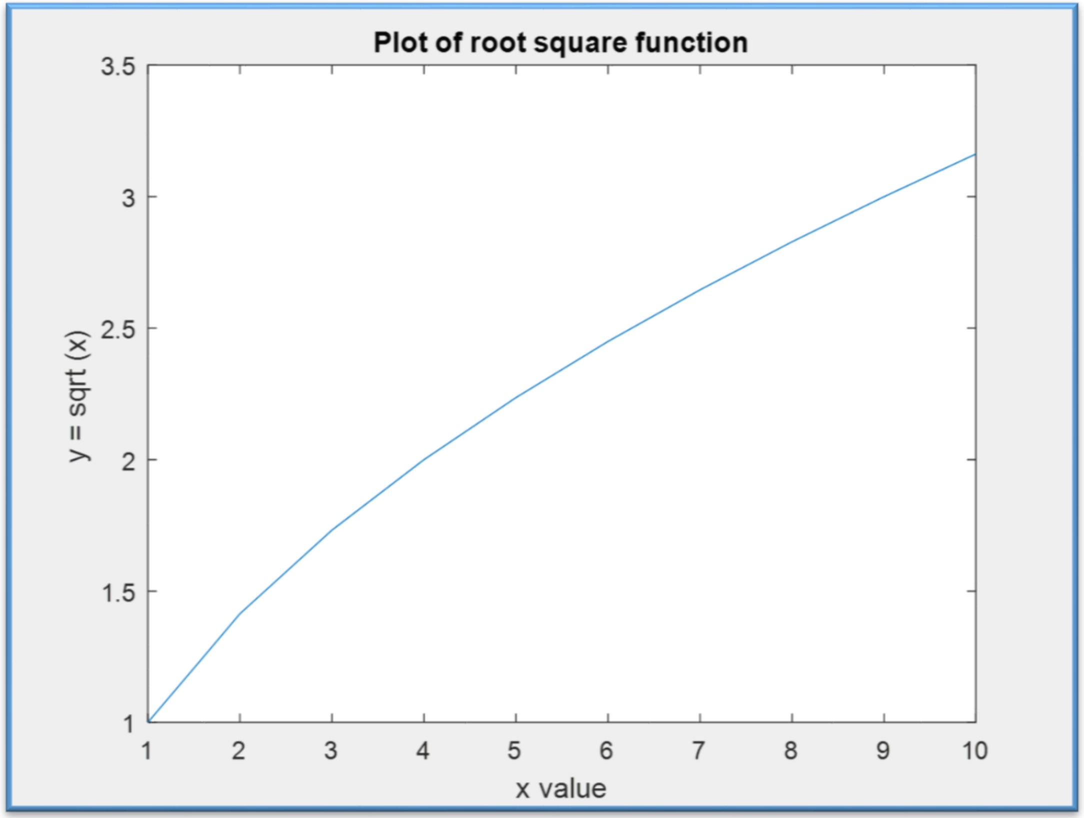
Creates a 2-D line plot of the data in Y versus the corresponding values in X.

The following example shows a script file introduced to plot the **root square** function in the interval [1 10]: $y = \sqrt{x}$

Let us modify our script by adding **title** and **labels** to axes.

**`xlabel()`** and **`ylabel()`**: used to place the *x* label and *y* label in the graph respectively.
**`title()`**: used to show the title of the graph in the plot.

```
>> X = 1:10;

>> Y = sqrt(X);

>> figure;

>> plot(X, Y);

>> xlabel('x value');

>> ylabel('y = sqrt (x)’);

>> title('Plot of root square function');
```



# 2. 2D plot

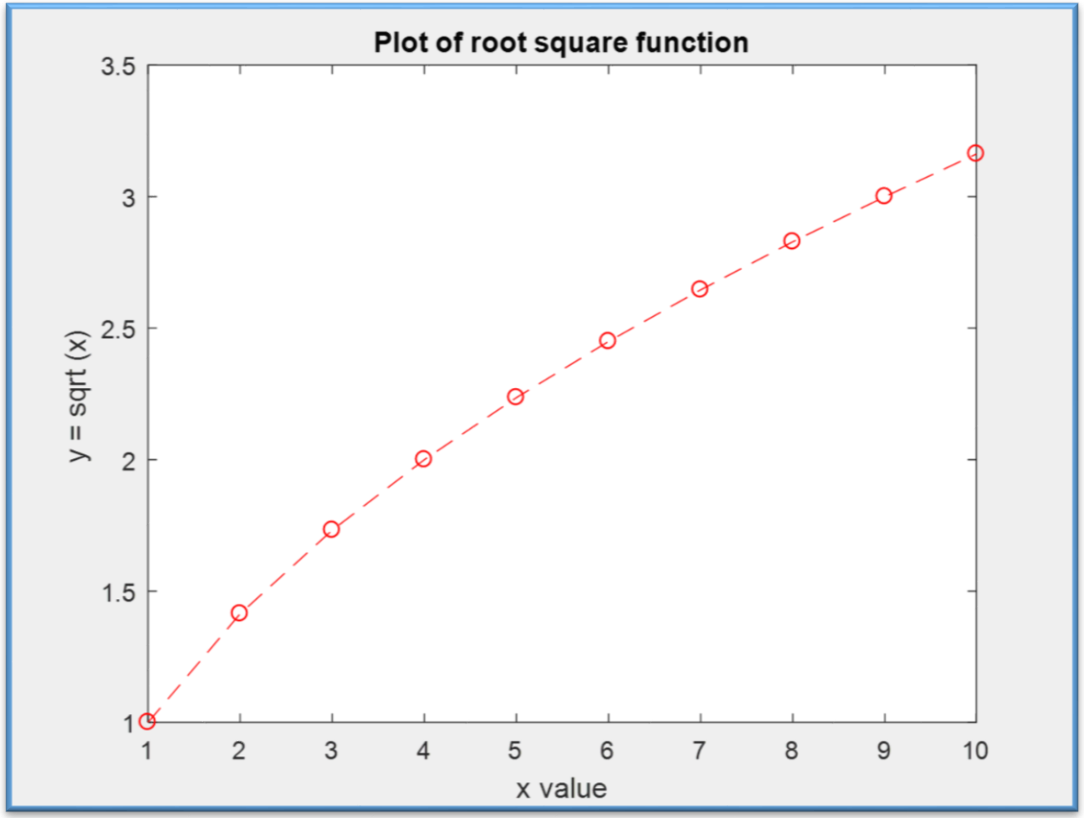
More specifications can be added to the plot command :
`plot(X,Y,LineSpecifiction)` where

`Linespecification` sets the **line style**, **marker symbol**, and **color** which are specified as a vector of symbols `('xxx')`.

The symbols can be **placed in any order** and we do **not need to specify** all three characteristics.

In this example, the line is specified by 'r--o’
‘r’ : indicates the color of the line (red),
‘--‘ indicates the type of the line (--------)
‘o’ represents the marker of the points.

```
>> X = 1:10;

>> Y = sqrt(X);

>> figure;

>> plot(X, Y, 'r--o');

>> xlabel('x value');

>> ylabel('y = sqrt (x)’);

>> title('Plot of root square function');
```



# 3. Plotting Multi Graph

To plot multiple lines in the same figure using the same axes we use the notation indicated below:

`plot(X1,Y1,...,Xn,Yn)` : Creates n 2-D line plots of the data in `Yi` versus the corresponding values in `Xi`.

The next example plots three curves of three functions:$\sqrt{x}$, $2\sqrt{x}$ and $3\sqrt{x}$ in the interval [1 10].

We add **more specifications** to the corresponding lines and a **legend** to differentiate between the plotted curves.

```
>> X = 1:10;

>> Y1 = sqrt(X);

>> Y2 = 2 * sqrt(X);

>> Y3 = 3 * sqrt(X);

>> figure

>> plot(X, Y1, X, Y2, X, Y3);

>> xlabel('x values');

>> ylabel('y = sqrt (x)');

>> title('Plot of square root function');
```

# 3. Plotting Multip Graph

To plot multiple lines in the same figure using the same axes we use the notation indicated below:

**plot(X1,Y1,...,Xn,Yn)** : Creates n 2-D line plots of the data in **Yi** versus the corresponding values in **Xi**.

The next example plots three curves of three functions:$\sqrt{x}$, $2\sqrt{x}$ and $3\sqrt{x}$ in the interval [1 10].

We add **more specifications** to the corresponding lines and a **legend** to differentiate between the plotted curves.

```
>> X = 1:10;

>> Y1 = sqrt(X);

>> Y2 = 2 * sqrt(X);

>> Y3 = 3 * sqrt(X);

>> figure

>> plot(X, Y1,'r', X, Y2,'go', X, Y3, 'b--*');

>> xlabel('x values');

>> ylabel('y = sqrt (x)');

>> title('Plot of square root function');

>> legend(' y1', ' y2',' y3');
```

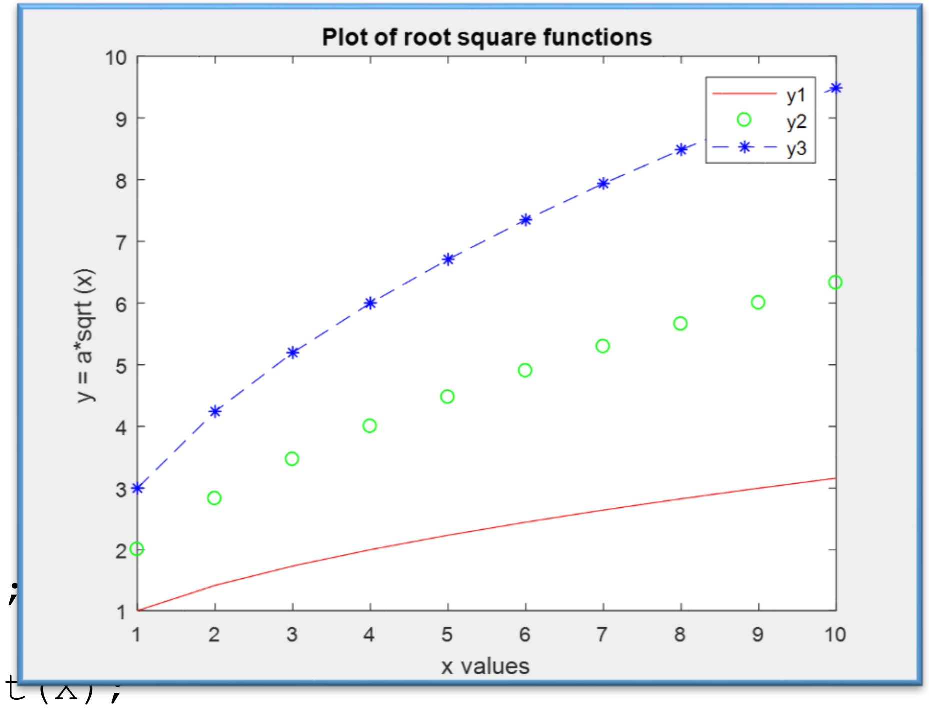
# 3. Plotting Multi Graph

To plot multiple lines in the same figure using the same axes we use the notation indicated below:

**`plot(X1,Y1,...,Xn,Yn)`** : Creates n 2-D line plots of the data in **`Yi`** versus the corresponding values in **`Xi`**.

The next example plots three curves of three functions:$\sqrt{x}$, $2\sqrt{x}$ and $3\sqrt{x}$ in the interval [1 10].

We add **more specifications** to the corresponding lines and a **legend** to differentiate between the plotted curves.



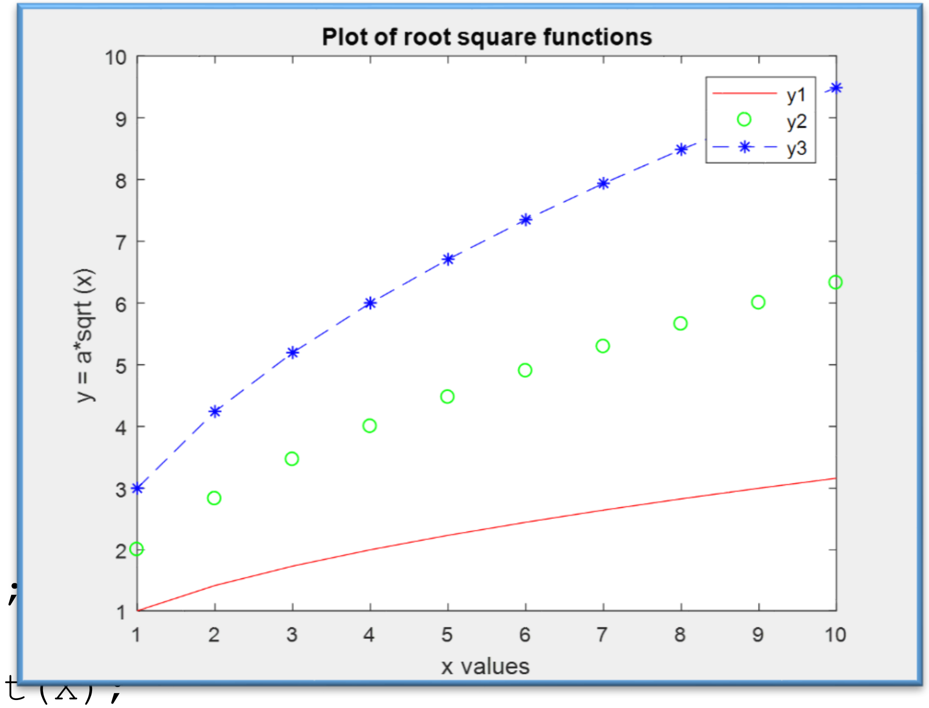
```
>> X = 1:10;

>> Y1 = sqrt(X);

>> Y2 = 2 * sqrt(X);

>> Y3 = 3 * sqrt(X);

>> figure

>> plot(X, Y1,'r', X, Y2,'go', X, Y3, 'b--*');

>> xlabel('x values');

>> ylabel('y = sqrt (x)');

>> title('Plot of square root function');

>> legend(' y1', ' y2',' y3’);
```

# 3. Plotting Multi Graph

In this script, we have added the following specifications to the lines of the corresponding curves:

- **'r'**: use a red line with no markers for the first curve;
- **'go'** : use only green circles markers for the second curve;
- **'b--*'** : use a blue dashed line with star markers for the third curve ;

The command **legend** is introduced to add a legend to the curve



```
>> X = 1:10;

>> Y1 = sqrt(X);

>> Y2 = 2 * sqrt(X);

>> Y3 = 3 * sqrt(X);

>> figure

>> plot(X, Y1,'r', X, Y2,'go', X, Y3, 'b--*');

>> xlabel('x values');

>> ylabel('y = sqrt (x)');

>> title('Plot of square root function');

>> legend(' y1', ' y2',' y3’);
```

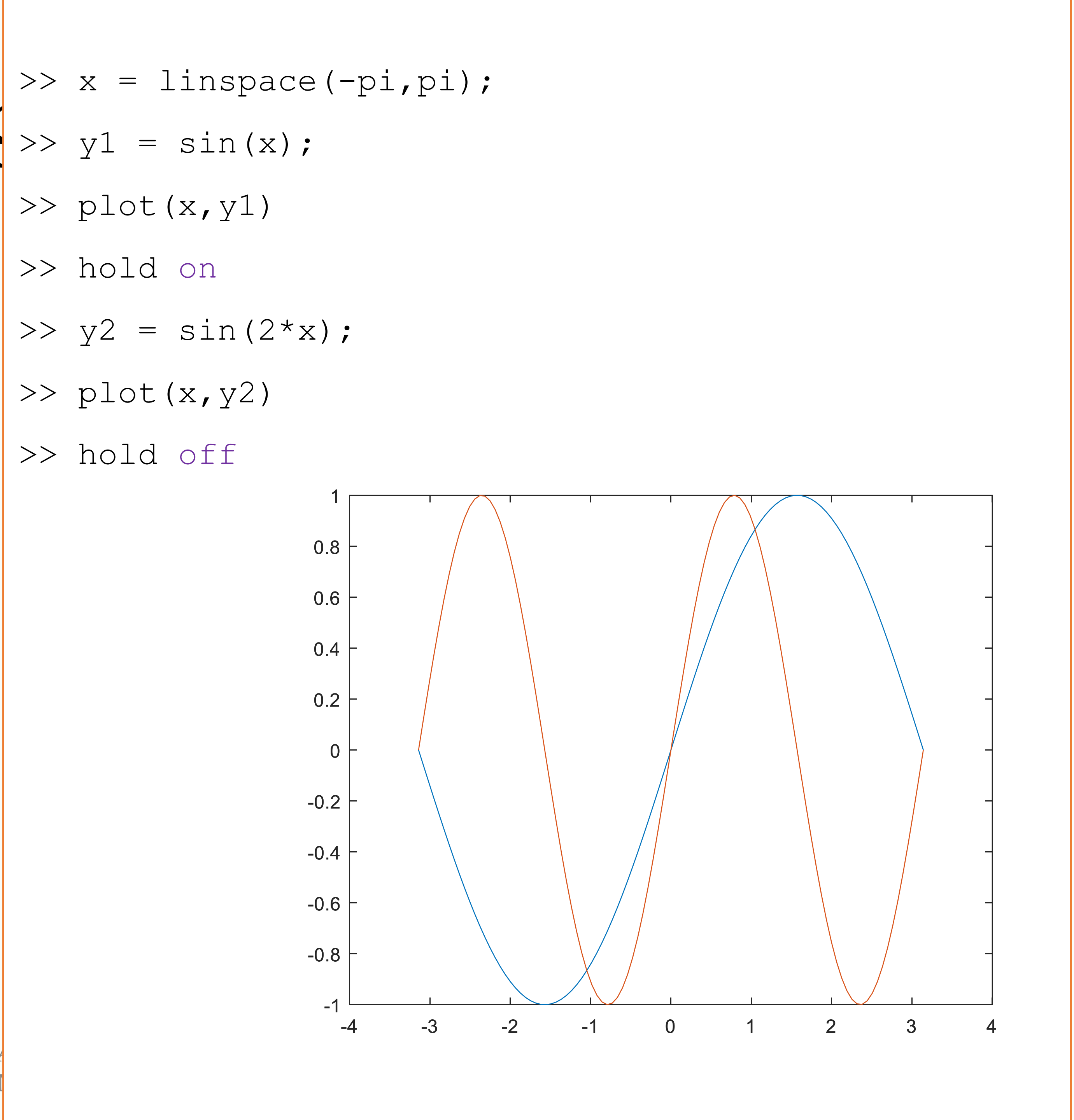
# 3. Plotting Multi Graph

To add new plots to existing one, we use the command

**`hold on`**

To turn off the addition of the plots in same graph we use the notation

**`hold off`**



```
>> x = linspace(-pi,pi);
>> y1 = sin(x);
>> plot(x,y1)
>> hold on
>> y2 = sin(2*x);
>> plot(x,y2)
>> hold off
```

# 4. Plotting Multiple Plots in Separate views

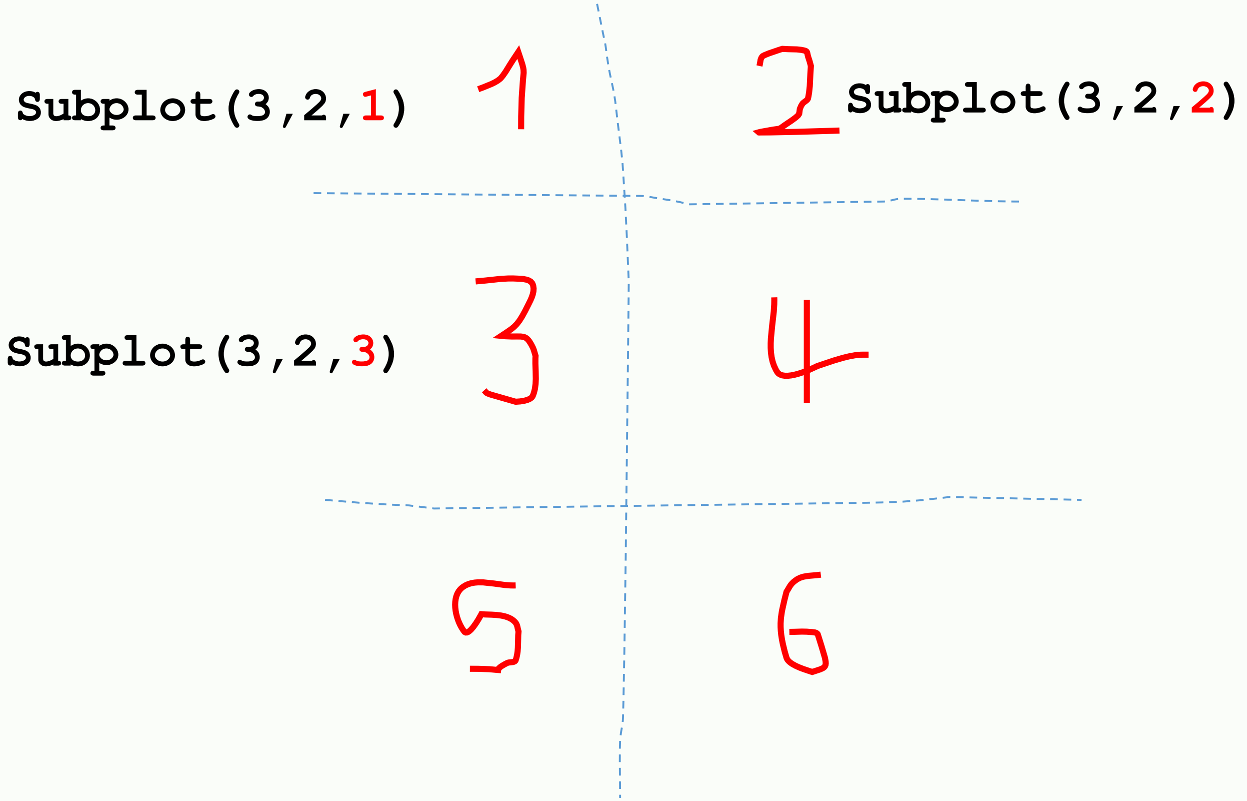
To show in a separate view multiple graphs, we use the **`subplot()`** command defined by the notation below.

**`subplot(row,coloumn,position)`**

This command creates a plot with **row-by-column** grid.

**`position`** represents the position of the subplot in the grid, where the subplot positions are numbered by rows.

The below figure represents the subplot positions on **3 by 2** grid.



**`Subplot(3,2,1)`**

**`Subplot(3,2,2)`**

**`Subplot(3,2,3)`**

3 by 2 grid

# 4. Plotting Multi views

To show in a separate view multiple graphs, we use the **`subplot()`** command defined by the notation below.

**`subplot(row,coloumn,position)`**

This command creates a plot with **row-by-column** grid.

**`position`** represents the position of the subplot in the grid, where the subplot positions are numbered by rows.

The below figure represents the subplot positions on **3 by 2** grid.

```
>> X = linspace(0,10);

>> Y1 = sqrt(X);

>> Y2 = sin(X);

>> figure

>> subplot(2,1,1);

>> plot(X, Y1);

>> title('SubPlot 1 : y1’);

>> subplot(2,1,2);

>> plot(X, Y2);

>> title('SubPlot 2 : y2');
```

# 4. Plotting Multi views

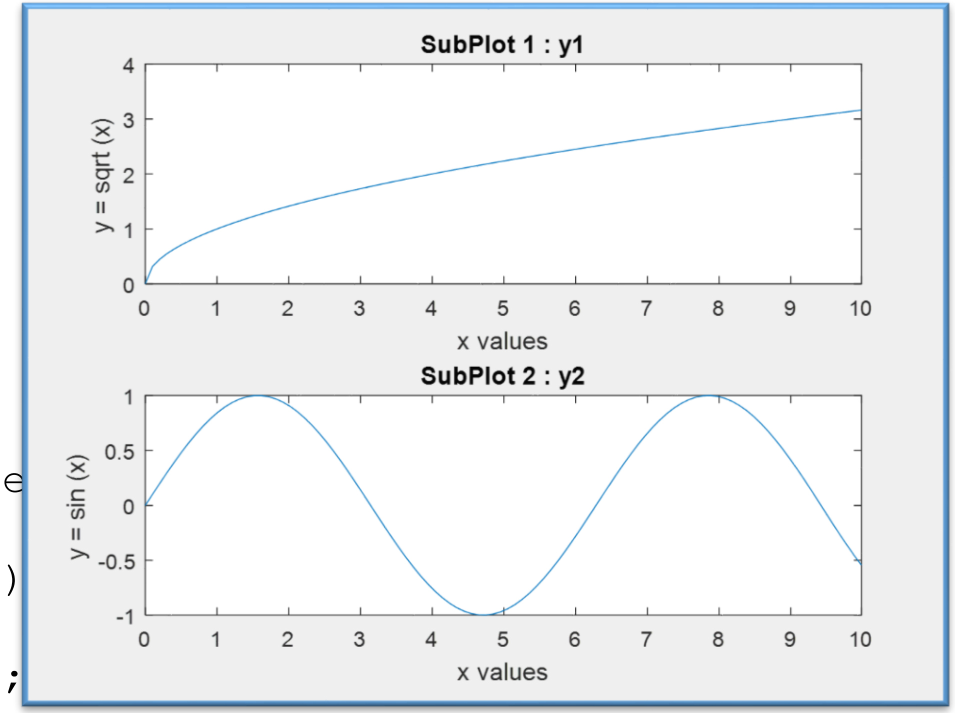
To show in a separate view multiple graphs, we use the **subplot()** command defined by the notation below.

**subplot(row,coloumn,position)**

This command creates a plot with **row-by-column** grid.

**position** represents the position of the subplot in the grid, where the subplot positions are numbered by rows.

The below figure represents the subplot positions on **3 by 2** grid.



```
>> X = linspace
>> Y1 = sqrt(X)
>> Y2 = sin(X);
>> figure
>> subplot(2,1,1);
>> plot(X, Y1);
>> title('SubPlot 1 : y1');
>> subplot(2,1,2);
>> plot(X, Y2);
>> title('SubPlot 2 : y2');
```

# Practice