Drawing Functions in MATLAB

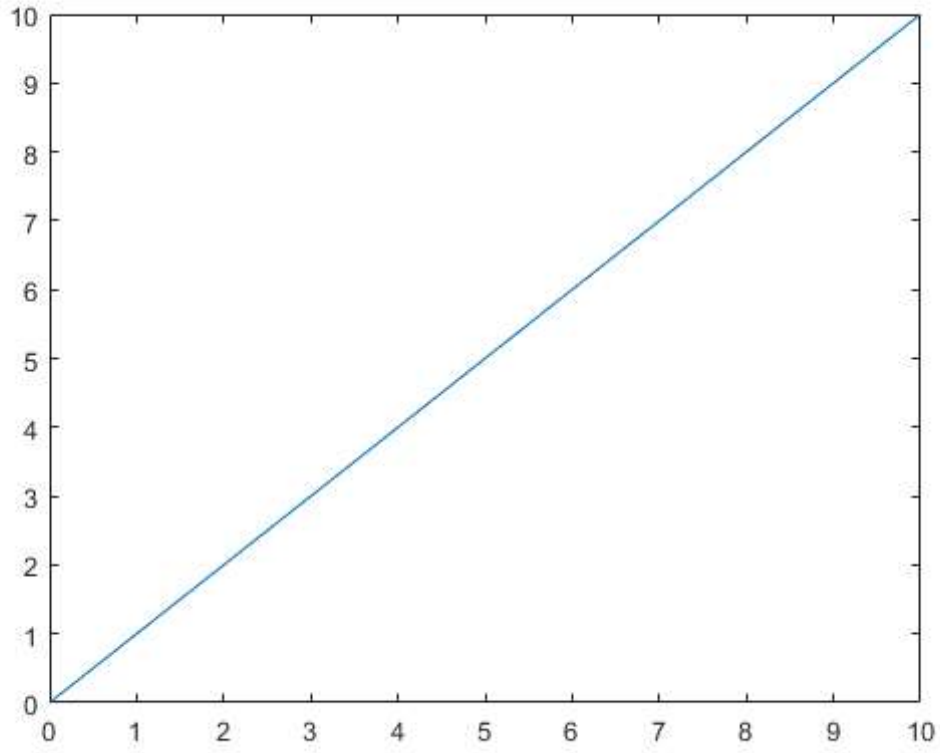
1. Function `plot`

Function `plot` is used to draw a linear coordinate graph on a 2D plane. To provide a set of x coordinates and corresponding y coordinates, a 2D curve with x and y as the horizontal and vertical coordinates can be drawn.

For example, draw function $y = x$, where, $0 \leq x \leq 10$

In [11]:

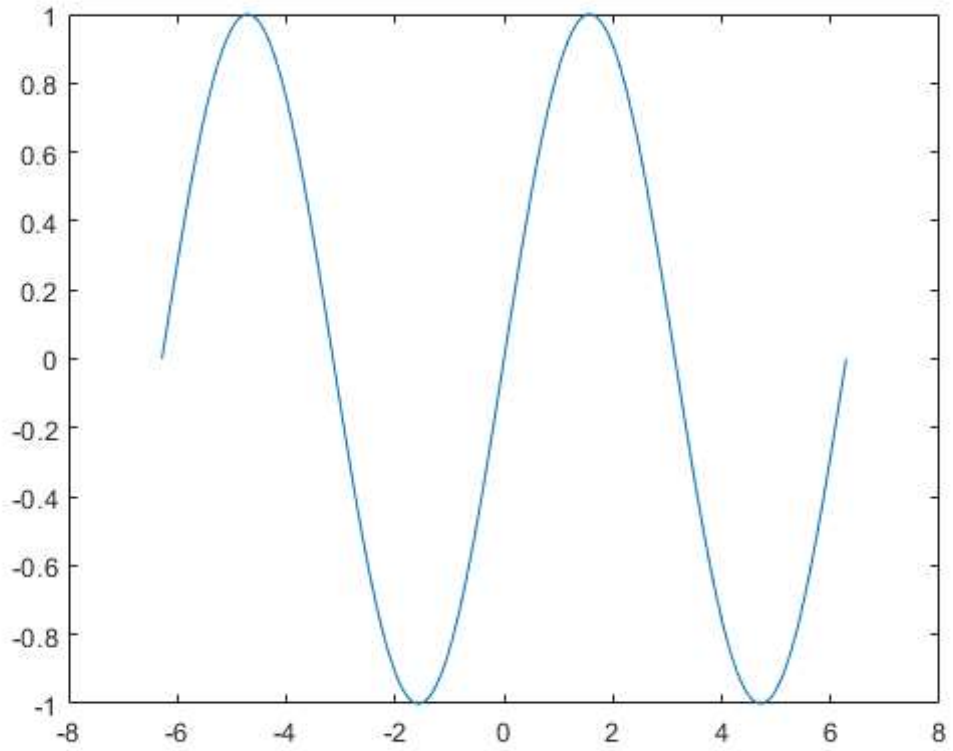
```
x = 0:10;
y = x;
plot(x, y)
```



Another example, draw function $y = \sin(x)$, where, $-2\pi \leq x \leq 2\pi$

In [6]:

```
x = -2*pi : pi/100 : 2*pi;
y = sin(x);
plot(x, y)
```



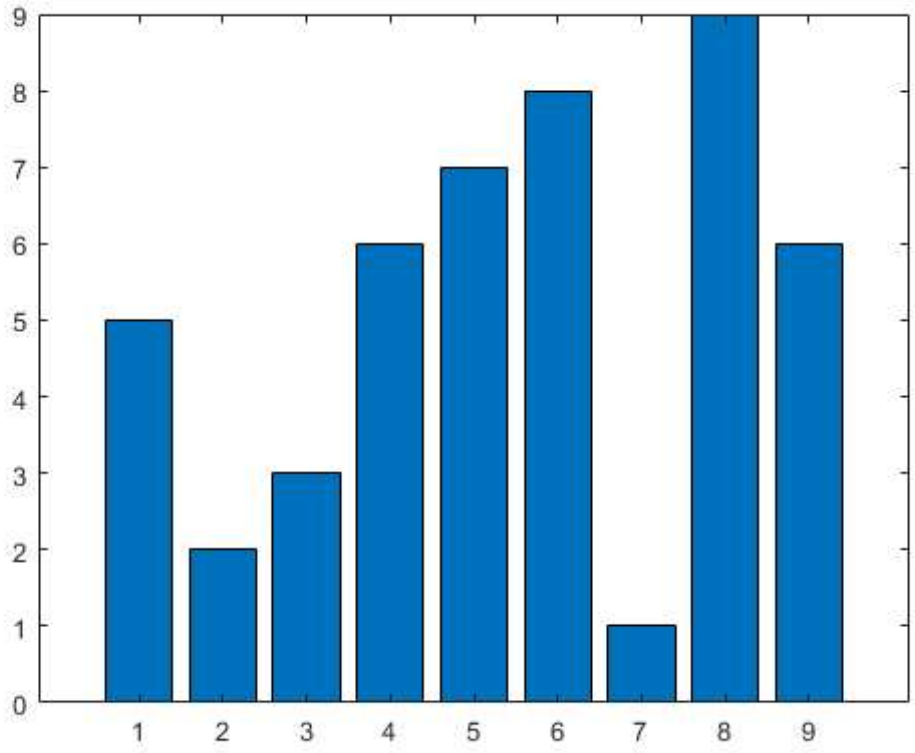
2. Function `bar`

Function `bar` is used to draw a histogram, which is very intuitive and practical for statistics or data acquisition.

For example, here are some random data, we can use function `bar` to draw it as a histogram. Here, function `bar` has only one parameter.

In [40]:

```
x = [5, 2, 3, 6, 7, 8, 1, 9, 6];
bar(x)
```



Sometimes we need to customize the data range of the x-axis, here we use

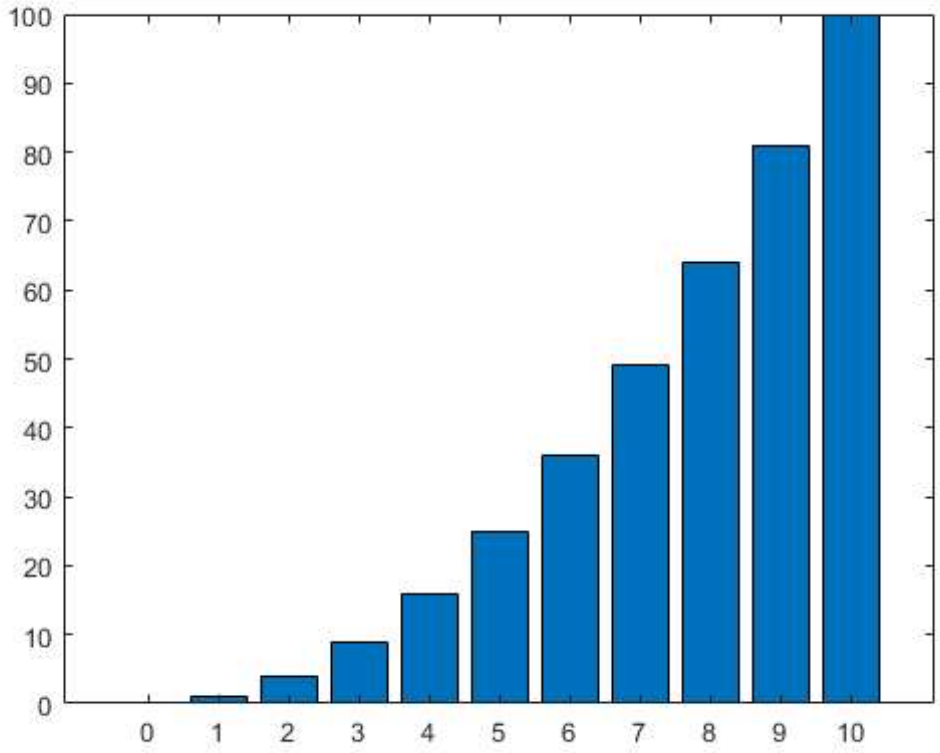
In [ ]:

```
bar(x, y)
```

where, x is the range of x-axis, it's a row vector and it must be monotonically increasing or decreasing. y is the data to show. For example,

In [44]:

```
x = 0:10;
y = x .* x;
bar(x, y)
```



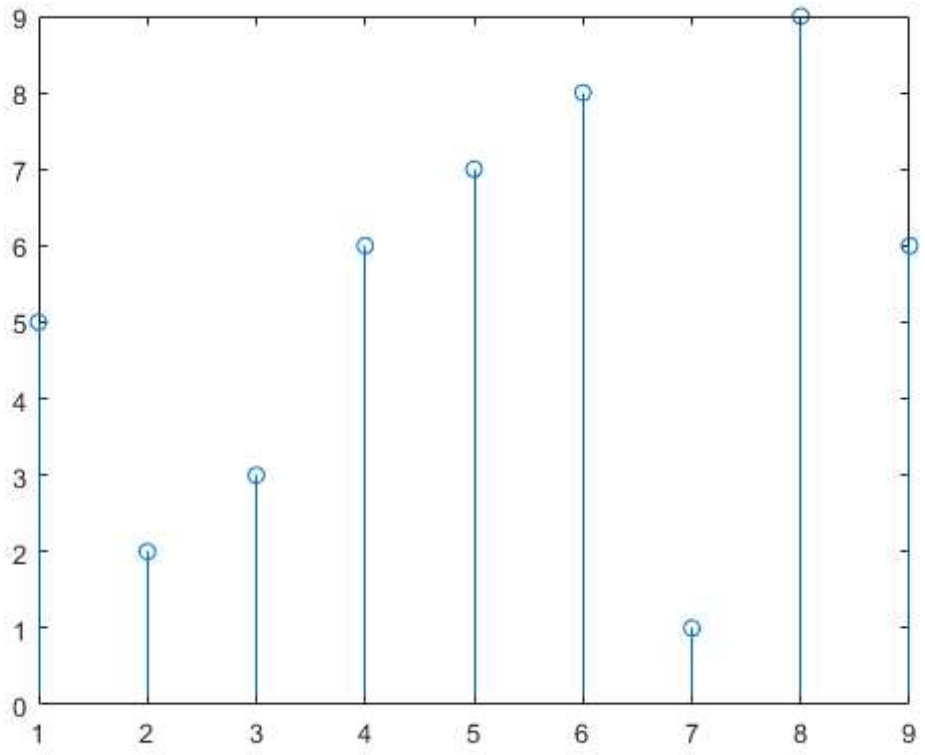
You can see the difference between `plot(x, y)` **and** `bar(x, y)`

3. Function `stem`

Function `stem` **is used to draw discrete data, which has the same syntax as the function** `bar` **, for example**

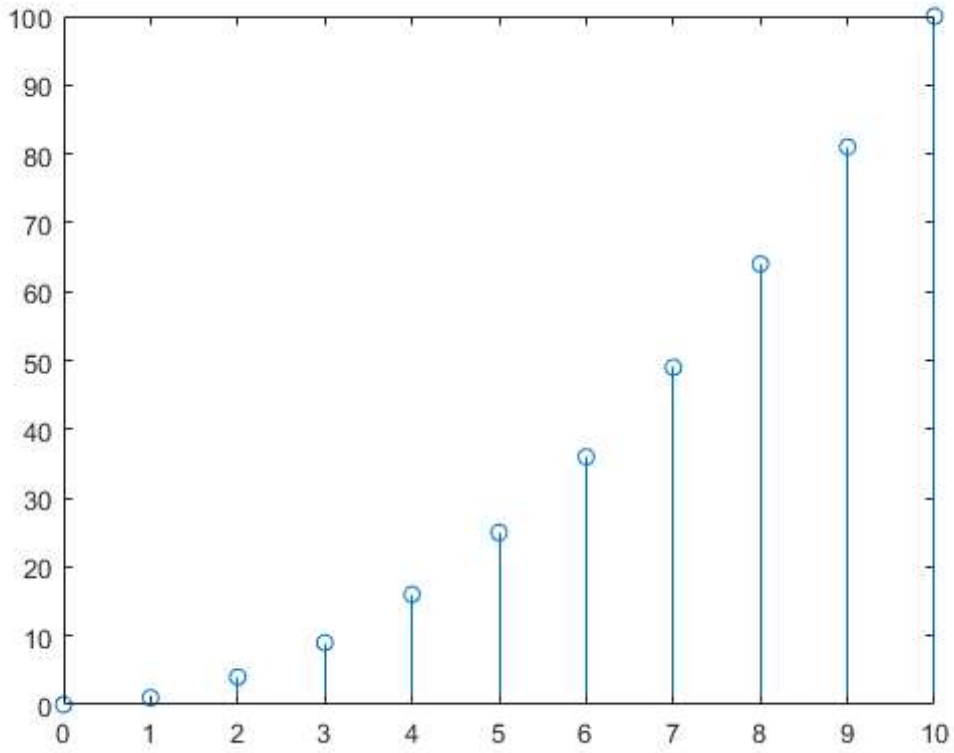
In [45]:

```
x = [5, 2, 3, 6, 7, 8, 1, 9, 6];
stem(x)
```



In [47]:

```
x = 0:10;
y = x .* x;
stem(x, y)
```



4. Function `subplot`

In practical applications, it is often necessary to draw several independent graphics in a graphics window, which requires segmentation of the graphics window.

The segmented graphics window consists of several drawing areas, each of which can create an independent coordinate system and draw graphics.

MATLAB provides a `subplot` function to split the current window into several plots, each representing a separate subgraph and an independent coordinate system.

Function `subplot` has the basic syntax:

In [ ]:

```
subplot(m, n, p)
```

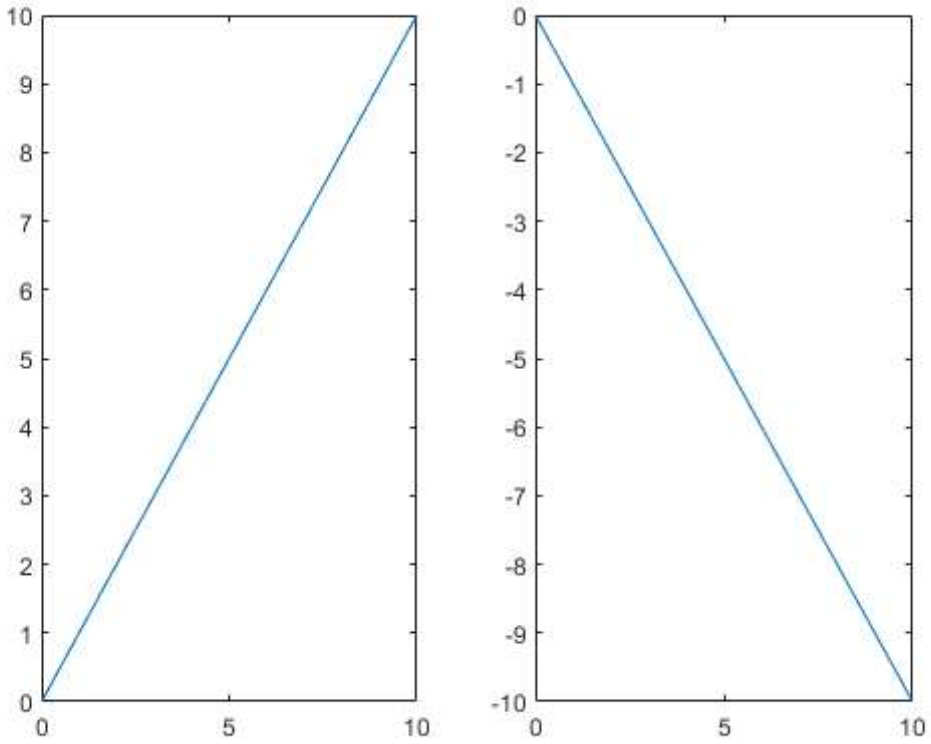
It divides the current window into $m \times n$ drawing areas, m lines, n drawing areas per line, and the area numbers are numbered first by row. The p-th area is the current active area. Each plot area allows graphs to be drawn separately in different coordinate systems.

For example, draw function $y = x$ and $y = -x$ in one figure, where, $0 \leq x \leq 10$

In [12]:

```
x = 0:10;
y1 = x;
y2 = -x;

subplot(1, 2, 1), plot(x, y1);
subplot(1, 2, 2), plot(x, y2);
```



5. Graphic Annotation

When drawing a graphic, you can add some descriptions to the graphic, such as the name of the graphic, the description of the axis, and the meaning of a part of the graphic.

Function `title` is used to title the image, which has the basic syntax

In [ ]:

```
title('title content')
```

Function `xlabel` and `ylabel` are used to add labels to the x and y axes.

In [ ]:

```
xlabel('x label content')
ylabel('y label content')
```

Function `legend` is used to draw the legend. Which has the basic syntax

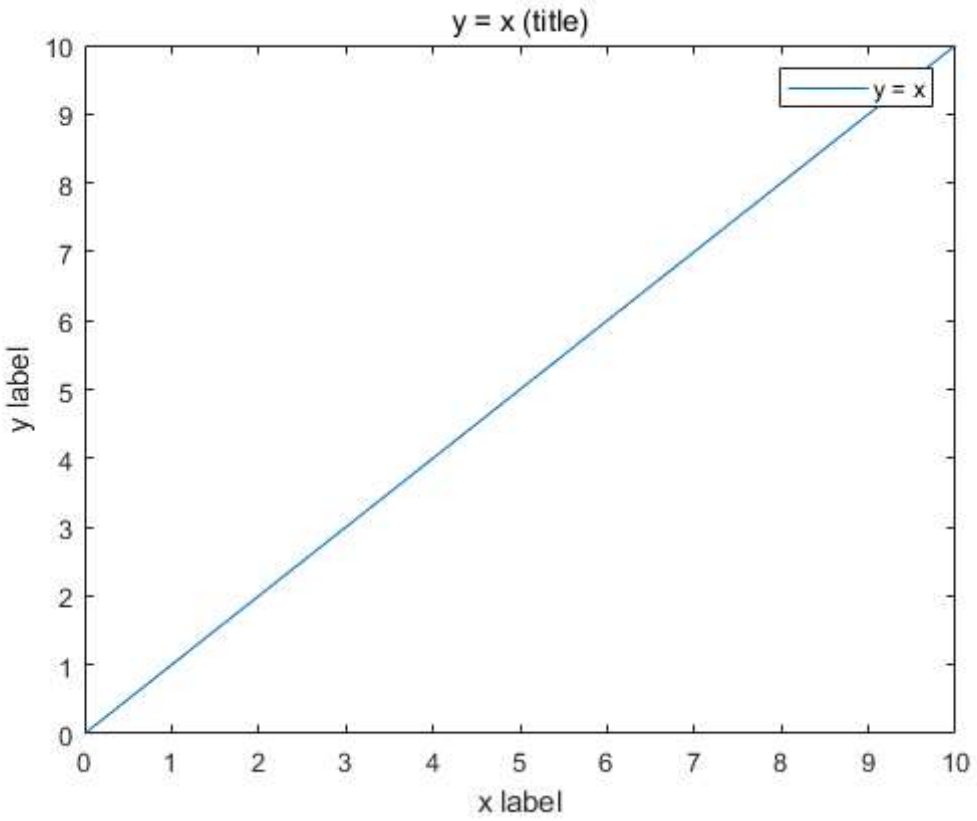
In [ ]:

```
legend('legend content')
```

For example, draw function $y = x$ with title, x-axis label, y-axis label and legend.

In [36]:

```
x = 0:10;
y = x;
plot(x, y)
title('y = x (title)')
xlabel('x label')
ylabel('y label')
legend('y = x')
```



Classwork

Draw the following two functions in one figure shown below. Each function image contains a title.

Function 1, $y = x^3 + x^2 + x$**, where** $-5 \leq x \leq 5$

Function 2, $y = \sin(x) + x$**, where** $0 \leq x \leq 20$

Note: the power of the vector requires the operator `.^`, for example `x.^3`.

In [2]:

```
x1 = -5:5;
y1 = x1.^3 + x1.^2 + x1;
subplot(2, 1, 1), plot(x1, y1), title('function 1');

x2 = 0:20;
y2 = sin(x2) + x2;
subplot(2, 1, 2), plot(x2, y2), title('function 2');
```