

subplot

Create axes in tiled positions

Syntax

<code>subplot(m,n,p)</code>	example
<code>subplot(m,n,p,'replace')</code>	example
<code>subplot(m,n,p,'align')</code>	
<code>subplot('Position',positionVector)</code>	example
<code>subplot(m,n,p,ax)</code>	example
<code>subplot(__ ,Name,Value)</code>	
 <code>h = subplot(__)</code>	example
 <code>subplot(h)</code>	example

Description

`subplot(m,n,p)` divides the current figure into an m-by-n grid and creates an axes for a subplot in the position specified by p. MATLAB® numbers its subplots by row, such that the first subplot is the first column of the first row, the second subplot is the second column of the first row, and so on. If the axes already exists, then the command `subplot(m,n,p)` makes the subplot in position p the current axes. [example](#)

`subplot(m,n,p,'replace')` deletes any existing axes in position p and creates a new axes. [example](#)

`subplot(m,n,p,'align')` creates a new axes so that the plot boxes are aligned. This is the default behavior.

`subplot('Position',positionVector)` creates a new axes at the position specified by positionVector. The positionVector is a four-element vector of the form [left,bottom,width,height], such that the entries are normalized values between 0.0 to 1.0. If the position vector specifies an axes that overlaps any previous axes, then the new axes replaces the existing ones. [example](#)

`subplot(m,n,p,ax)` converts the existing axes, ax, into a subplot axes in the same figure. [example](#)

`subplot(__ ,Name,Value)` specifies properties for the axes using any of the input argument combinations in the previous syntaxes and one or more Name,Value pair arguments.

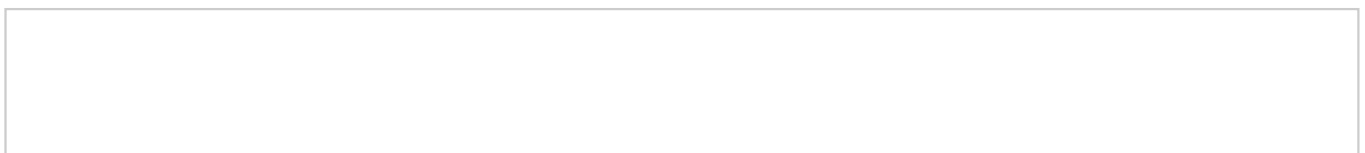
`h = subplot(__)` returns the axes object created by the subplot function. [example](#)

`subplot(h)` makes the axes with handle h the current axes of its figure, but does not make its figure the current figure if it is not already the current figure. [example](#)

Examples

Upper and Lower Subplots

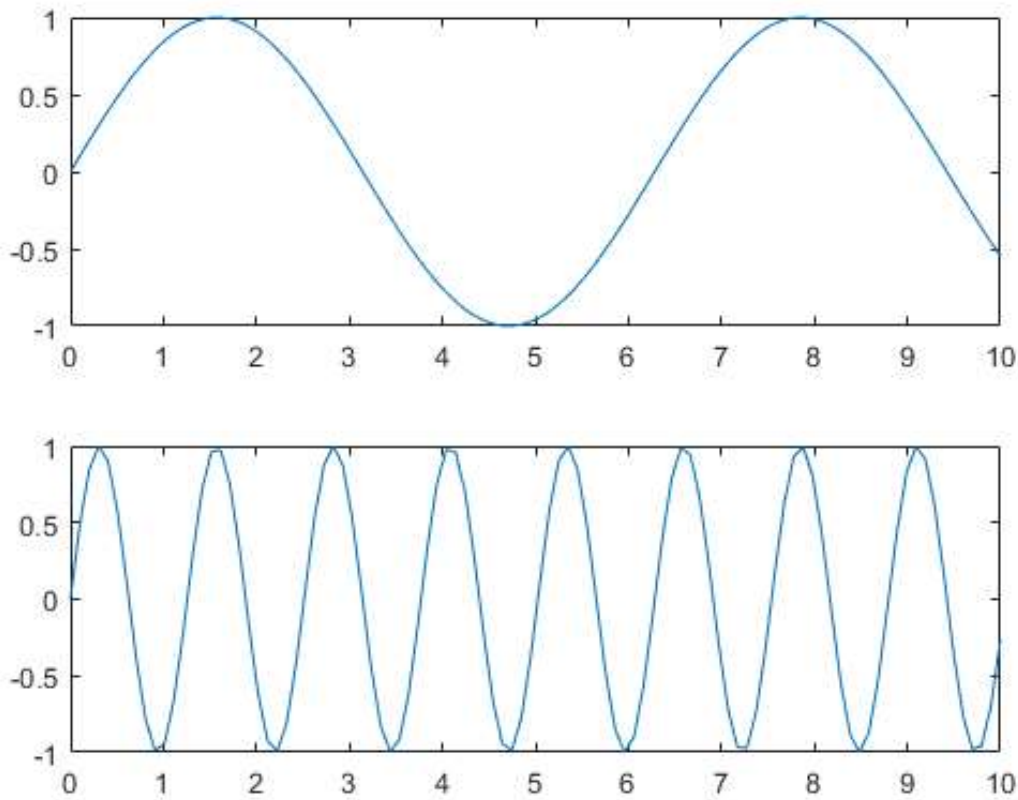
Create a figure with two stacked subplots. Plot a sine wave in each axes.



```
x = linspace(0,10);  
y1 = sin(x);  
y2 = sin(5*x);
```

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

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



Quadrant of Subplots

Create a figure divided into four subplots.

Define the data.

```
x = linspace(0,10);  
y1 = sin(x);  
y2 = sin(2*x);  
y3 = sin(4*x);  
y4 = sin(8*x);
```

Plot the four sine waves and title each subplot.

```

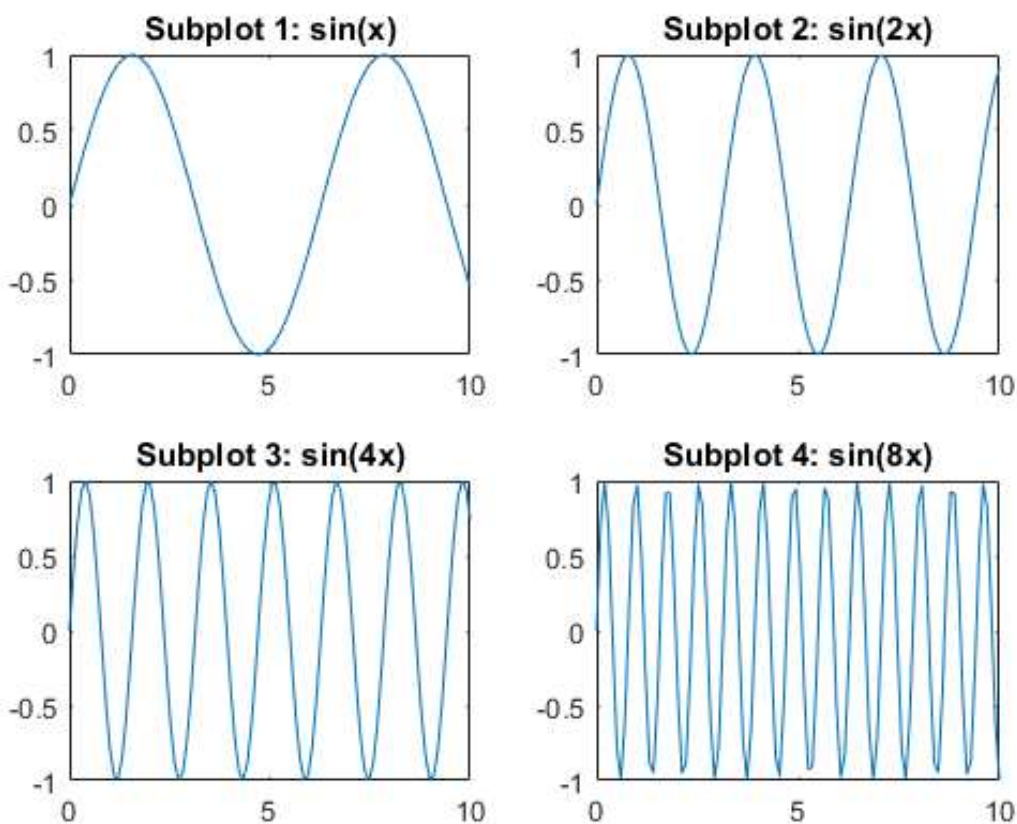
figure
subplot(2,2,1)
plot(x,y1)
title('Subplot 1: sin(x)')

subplot(2,2,2)
plot(x,y2)
title('Subplot 2: sin(2x)')

subplot(2,2,3)
plot(x,y3)
title('Subplot 3: sin(4x)')

subplot(2,2,4)
plot(x,y4)
title('Subplot 4: sin(8x)')

```



Subplots with Different Sizes

Create a figure containing subplots with different sizes.

Define a vector of sine values and a vector of polynomial values.

```

x = linspace(-3.8,3.8);
y_cos = cos(x);
y_poly = 1 - x.^2./2 + x.^4./24;

```

Plot the sine wave in the first subplot and the polynomial in the second subplot. Create a third subplot that spans the lower half of the figure and plot both vectors together. Add titles to each subplot.

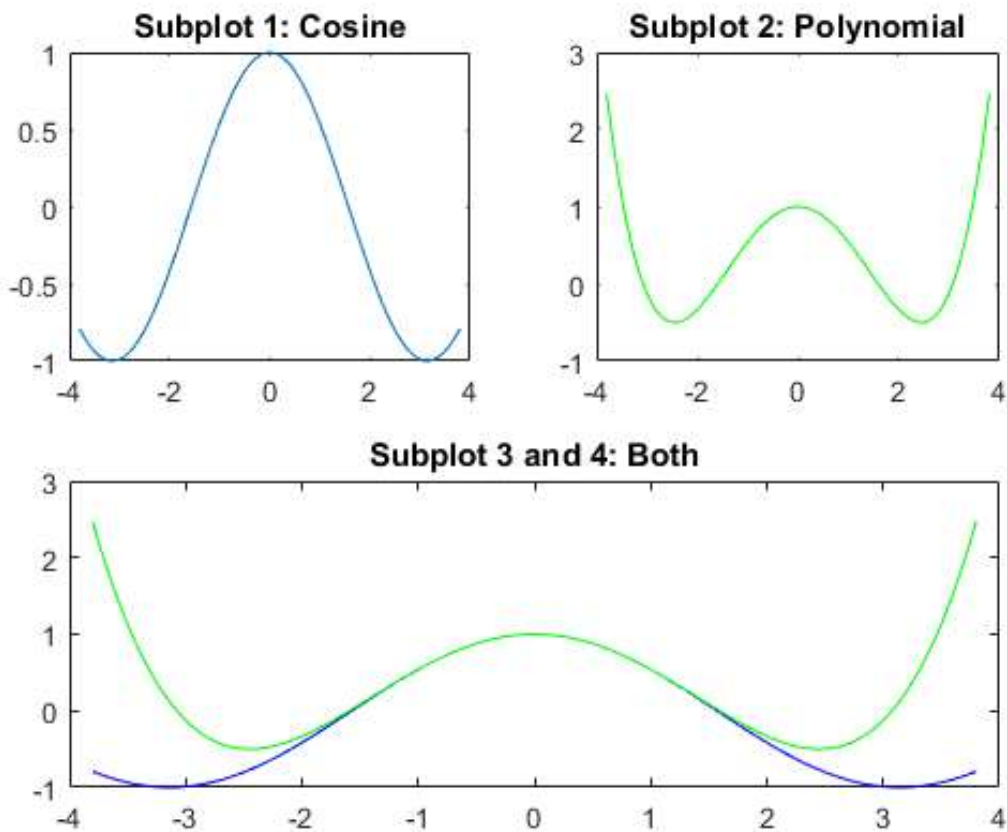
```

figure
subplot(2,2,1);
plot(x,y_cos);
title('Subplot 1: Cosine')

subplot(2,2,2);
plot(x,y_poly,'g');
title('Subplot 2: Polynomial')

subplot(2,2,[3,4]);
plot(x,y_cos,'b',x,y_poly,'g');
title('Subplot 3 and 4: Both')

```



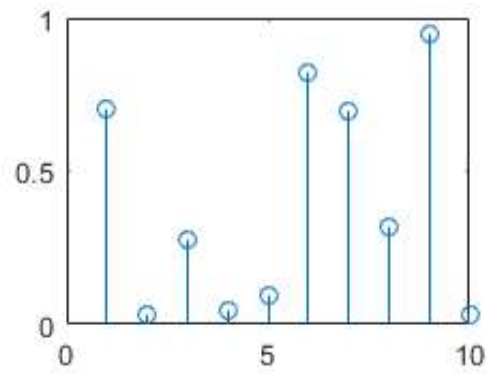
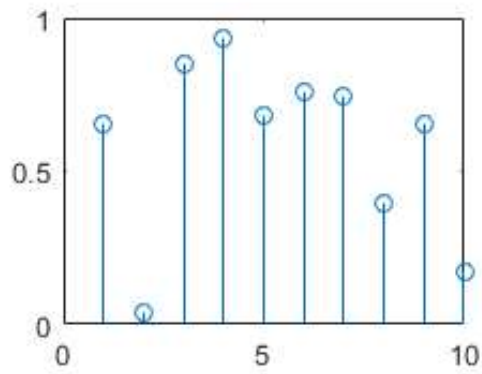
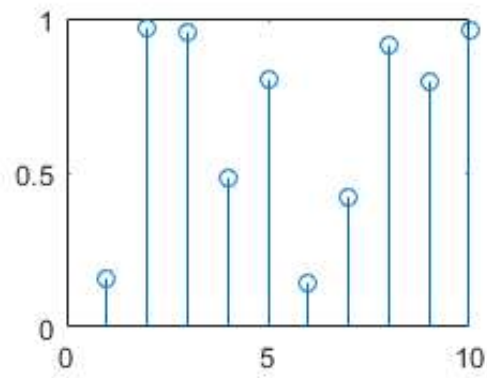
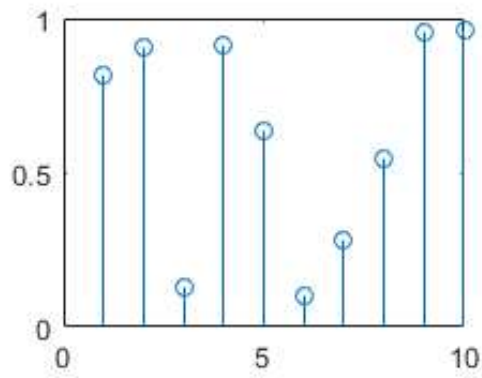
Replace Subplot with Empty Axes

Initialize the random number generator. Use a loop to create a figure with four stem plots of random data.

```

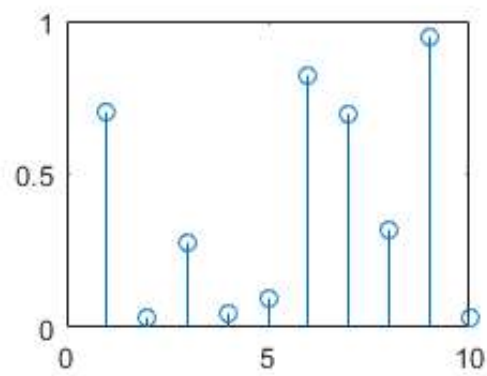
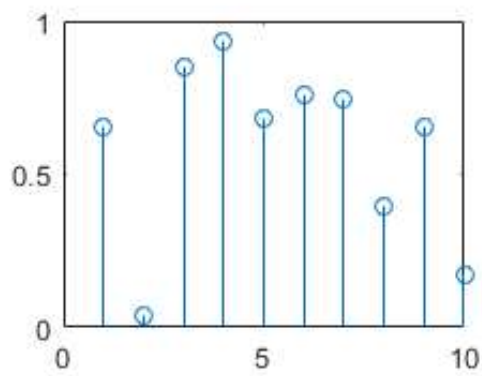
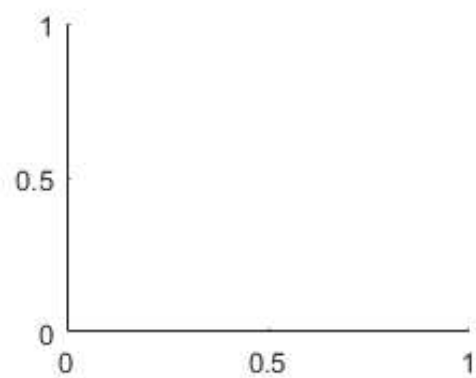
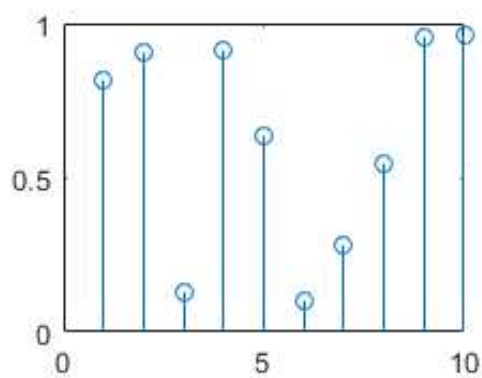
rng default
figure
for k = 1:4
    data = rand(1,10);
    subplot(2,2,k)
    stem(data)
end

```



Replace the second subplot with an empty axes.

```
subplot(2,2,2, 'replace')
```



Subplots at Specified Positions

Create a figure with two subplots that are not aligned.

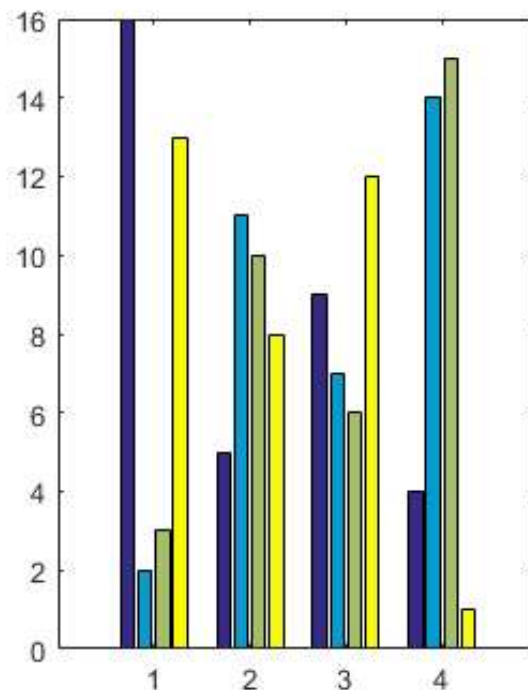
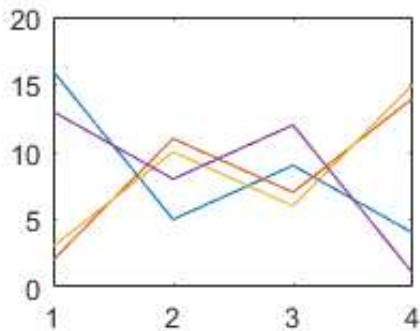
Define y as data from the magic function.

```
y = magic(4);
```

Plot y in one subplot. Create a bar graph of y in a second subplot. Specify a custom position for each subplot.

```
figure
positionVector1 = [0.1, 0.2, 0.3, 0.3];
subplot('Position',positionVector1)
plot(y)

positionVector2 = [0.5, 0.1, 0.4, 0.7];
subplot('Position',positionVector2)
bar(y)
```

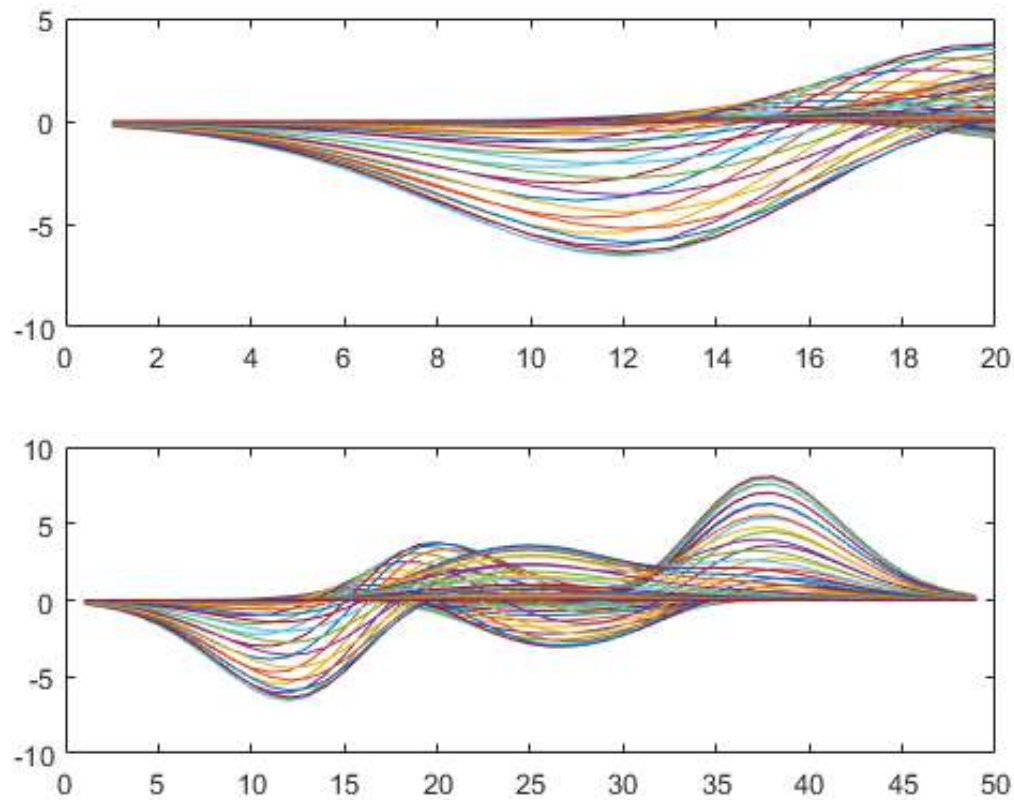


Return Subplot Axes Handle

Create a figure with two subplots. Plot the first 20 rows of the peaks function in the upper subplot. Plot the entire data set in the lower subplot.

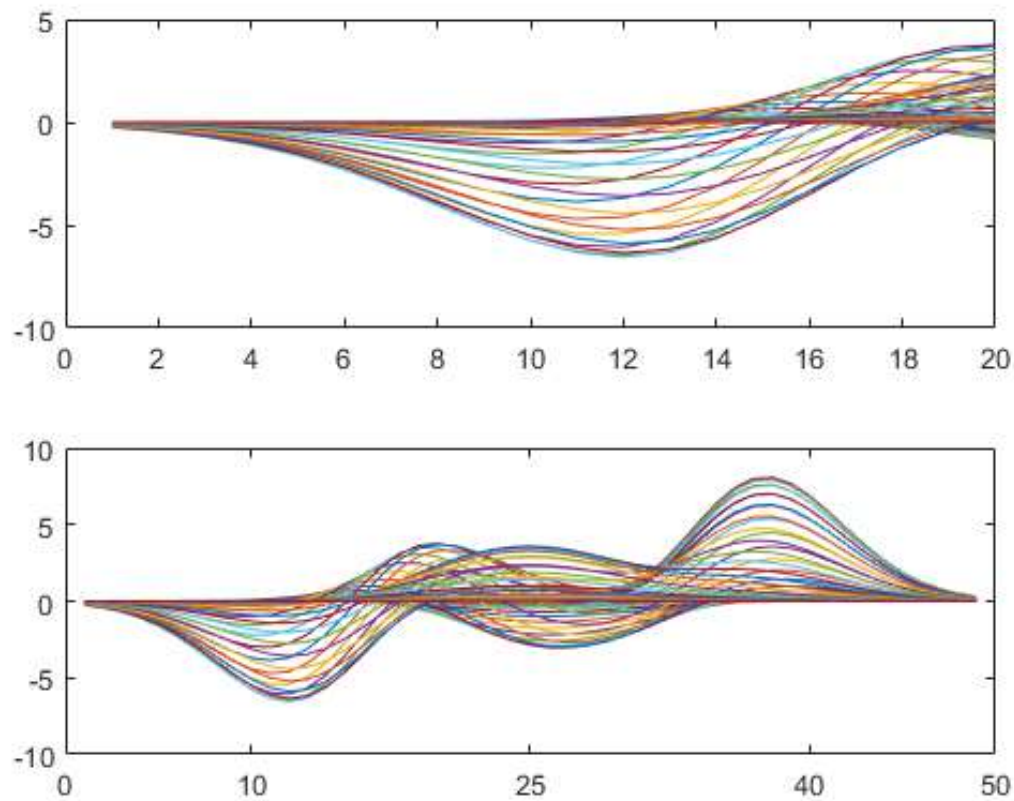
```
ax1 = subplot(2,1,1);
Z = peaks;
plot(ax1,Z(1:20,:))

ax2 = subplot(2,1,2);
plot(ax2,Z)
```



Change the tick marks for the lower subplot. Starting in R2014b, you can use dot notation to set properties. If you are using an earlier release, use the [set](#) function instead.

```
ax2.XTick = [0,10,25,40,50];
```



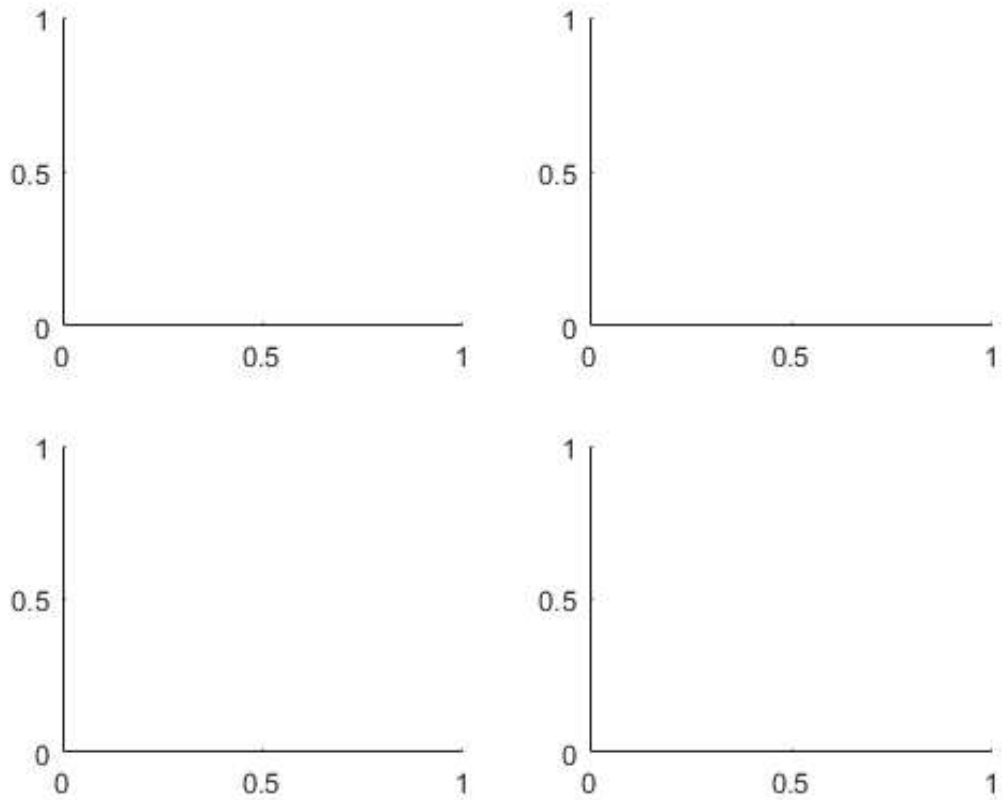
Some plotting functions set axes properties. Execute plotting functions before specifying axes properties to avoid overriding existing axes property settings.

Make Subplot Axes the Current Axes

Make a subplot the current axes using its axes handle.

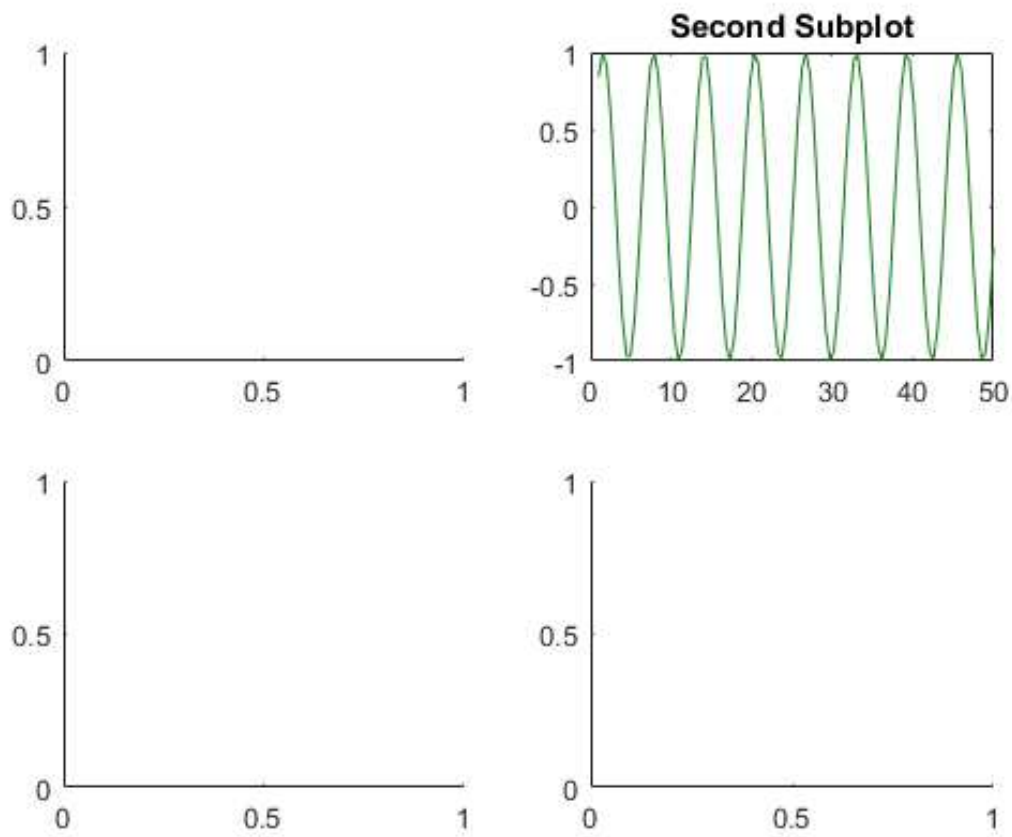
Create a figure with multiple subplots. Store the subplot axes handles in vector h.

```
figure;  
for k = 1:4  
    h(k) = subplot(2,2,k);  
end
```



Make the second subplot the current axes. Plot a sine wave and change the axis limits.

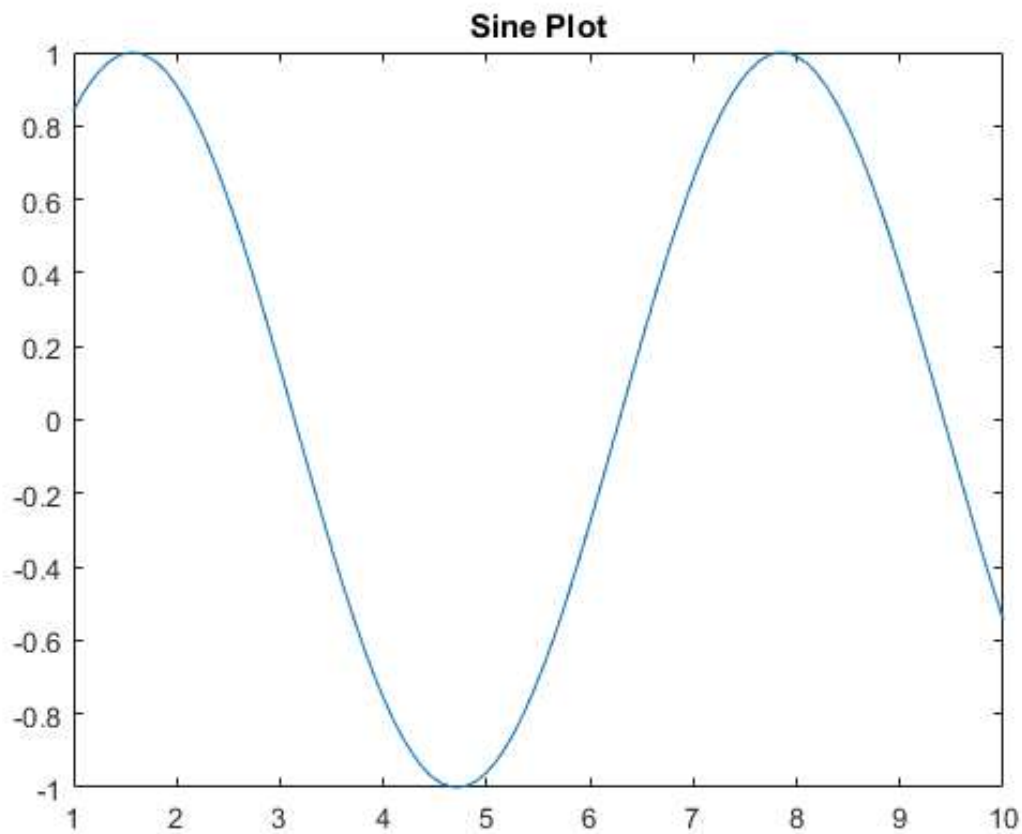
```
x = linspace(1,50);  
y = sin(x);  
  
subplot(h(2))  
plot(x,y,'Color',[0.1, 0.5, 0.1])  
title('Second Subplot')  
axis([0,50,-1,1])
```

Convert Existing Axes to Subplot

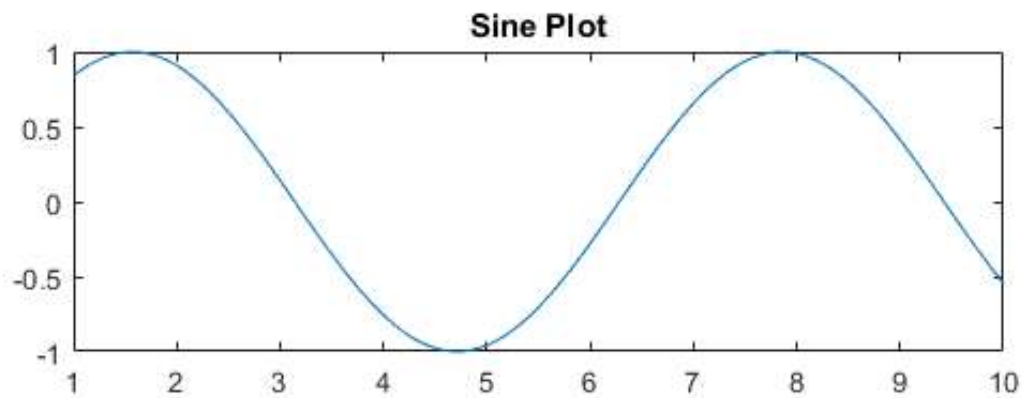
Create a plot with a title.

```
x = linspace(1,10);  
y = sin(x);  
plot(x,y)  
title('Sine Plot')
```



Get the current axes object using `gca`. Convert the current axes so that it is the lower subplot of the figure.

```
ax1 = gca;  
subplot(2,1,2,ax1)
```

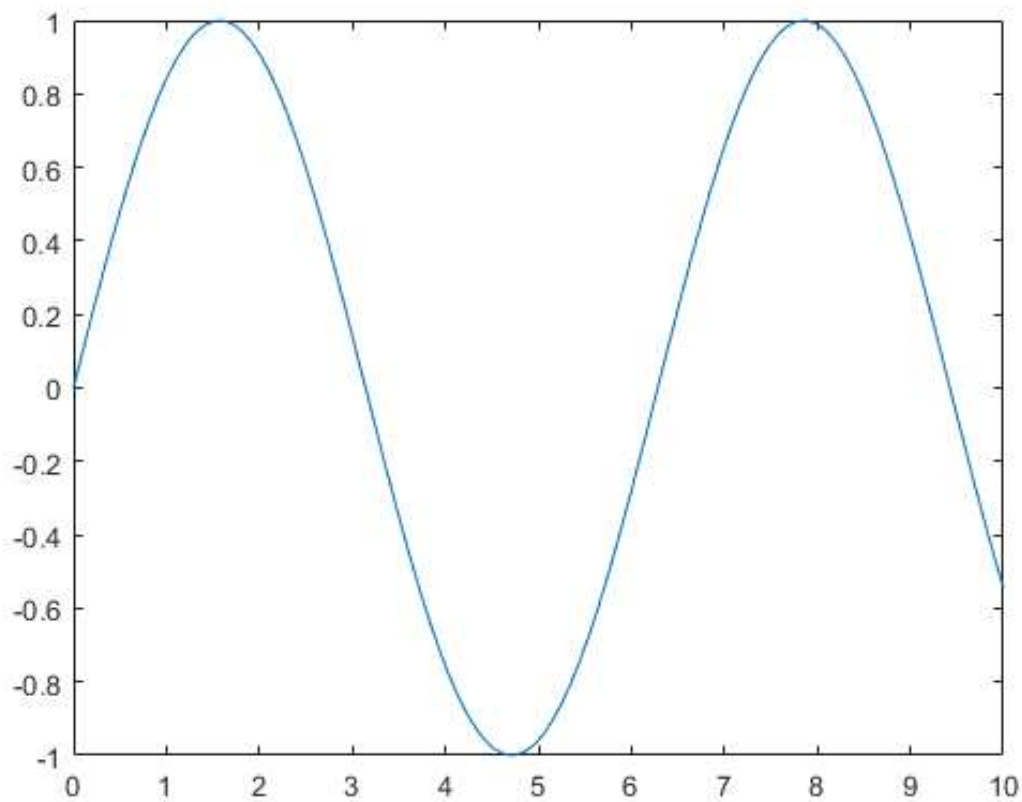


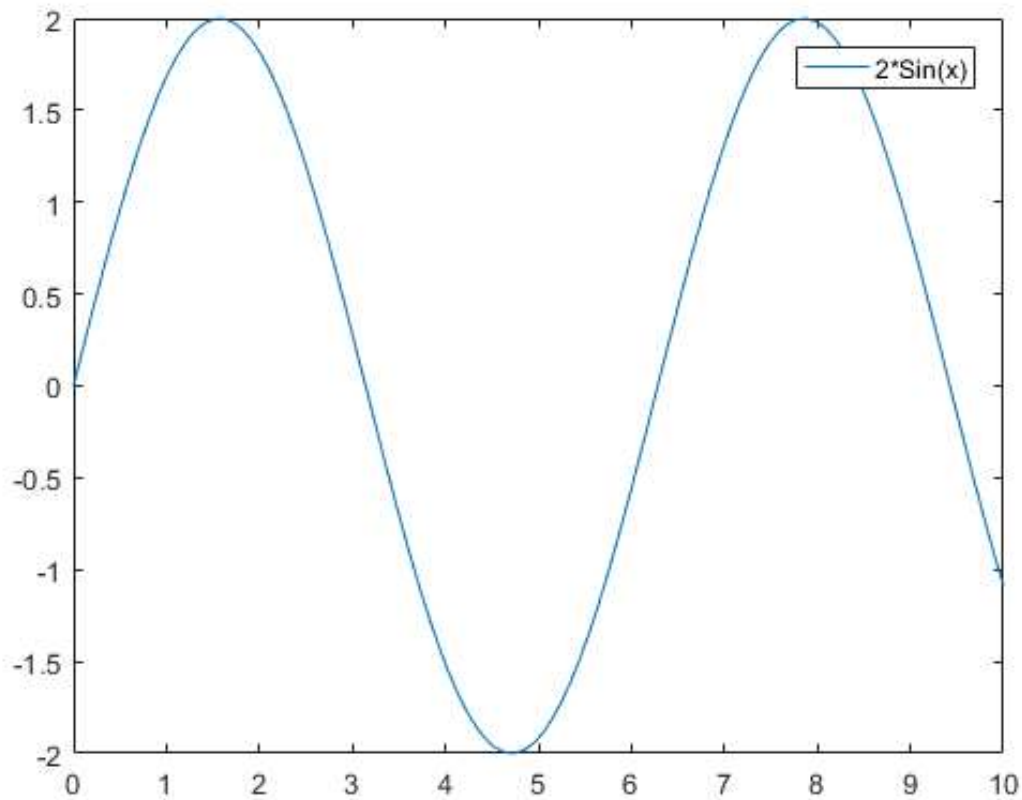
The `subplot` function uses the figure in which the original axes existed.

Combine Two Axes Into Single Figure With Subplots

Create two plots in two different figures and store the axes objects.

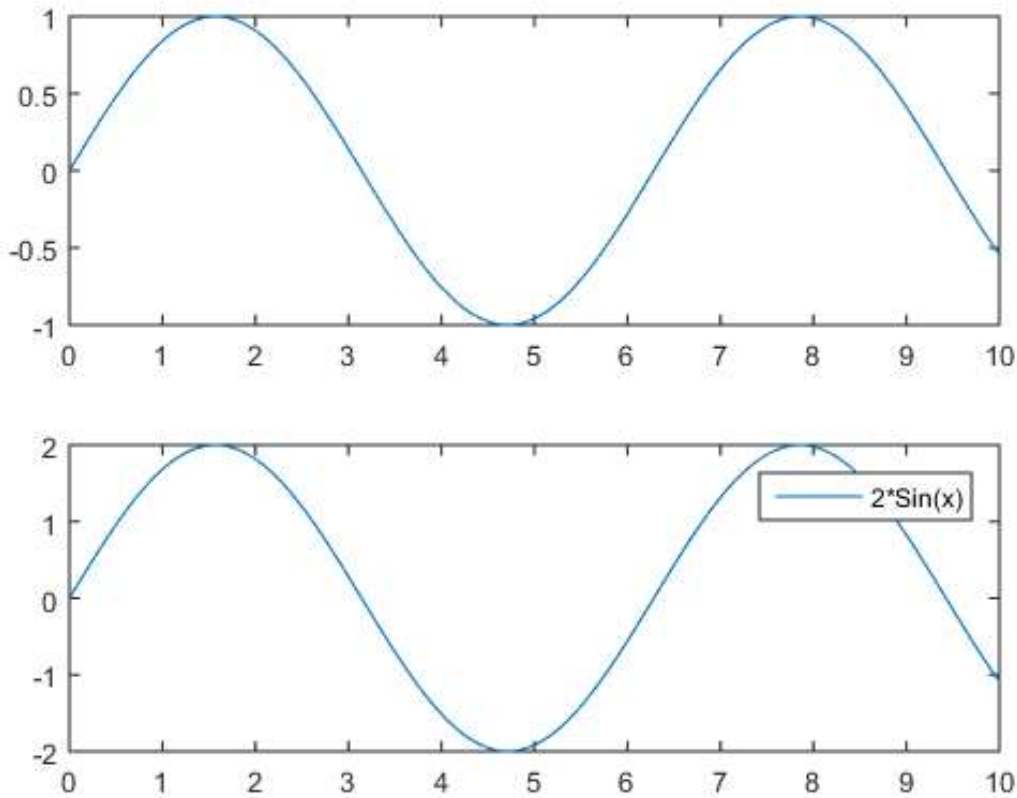
```
x = linspace(0,10);  
y1 = sin(x);  
figure  
plot(x,y1)  
ax1 = gca;  
  
y2 = 2*sin(x);  
figure  
plot(x,y2)  
leg = legend('2*Sin(x)');  
ax2 = gca;
```





Create copies of the two axes objects using `copyobj`. Set the parent of the copied axes to a third figure. Since legends and colorbars objects do not get copied with the associated axes, copy the legend with the axes.

```
f3 = figure;  
ax1_copy = copyobj(ax1,f3);  
subplot(2,1,1,ax1_copy)  
  
copies = copyobj([ax2,leg],f3);  
ax2_copy = copies(1);  
subplot(2,1,2,ax2_copy)
```



Input Arguments

[collapse all](#)



m — Number of grid rows
1 (default) | positive integer

Number of grid rows, specified as a positive integer.

Data Types: single | double



n — Number of grid columns
1 (default) | positive integer

Number of grid columns, specified as a positive integer.

Data Types: single | double



p — Grid position for new axes
positive integer | vector

Grid position for the new axes, specified as a positive integer or a vector of positive integers.

- If p is a positive integer, then `subplot` creates a new axes in grid position p .
- If p is a vector of positive integers, then `subplot` creates a new axes that spans the grid positions listed in p . For example, `subplot(2,3,[2,5])` creates one axes spanning positions 2 and 5. Use `subplot(2,3,[2,6])` to create one axes spanning positions 2, 3, 5, and 6.

Data Types: single | double



positionVector — Normalized position for new axes

four-element vector of values between 0.0 and 1.0

Normalized position for the new axes, specified as a four-element vector of values between 0.0 and 1.0 with the form `[left,bottom,width,height]`. The values are normalized with respect to the interior of the figure. The first two elements specify the position of the bottom-left corner of the axes in relation to the bottom-left corner of the figure. The last two elements specify the width and height of the new axes. Use this syntax to position an axes that does not align with grid positions.

Example: `[0.1, 0.1, 0.35, 0.35]`

Data Types: `single` | `double`



ax — Existing axes to convert to subplot

axes object | polar axes object

Existing axes to convert to a subplot, specified as an axes object or a polar axes object. To convert the current axes to a subplot, you can specify `ax` as `gca`, for example, `subplot(2,1,1,gca)`. To create an empty polar axes in a subplot position, specify `ax` as the [polaraxes](#) function, for example, `subplot(2,1,2,polaraxes)`.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name`, `Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside single quotes (`' '`). You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

Example: `'XGrid','on'`

You can set any axes properties for a subplot. Some plotting functions override existing axes property settings. Execute plotting functions before specifying axes properties to avoid overriding them. For a list of axes properties see [Axes Properties](#).

Output Arguments

[collapse all](#)



h — Axes object

scalar

Axes object, returned as a scalar. This is a unique identifier, which you can use to query and modify the properties of a specific axes.

More About

[collapse all](#)

▼ Tips

- `subplot(111)` is an exception and not identical in behavior to `subplot(1,1,1)`. For reasons of backwards compatibility, `subplot(111)` is a special case of `subplot` that does not immediately create an axes, but sets up the figure so that the next graphics command executes `clf reset`. The next

graphics command deletes all the figure children and creates a new axes in the default position. `subplot(111)` does not return a handle and an error occurs if code specifies a return argument.

- If a new subplot axes overlaps an existing axes, then MATLAB deletes the existing axes. To overlay a new axes on top of existing subplots, use the `axes` command. For example, `subplot('Position', [.35 .35 .3 .3])` deletes any underlying subplots, but `axes('Position', [.35 .35 .3 .3])` positions a new axes in the middle of the figure without deleting any underlying axes.
- When using a script to create subplots, MATLAB does not finalize the `Position` property value until either a `drawnow` command is issued or MATLAB returns to await a user command. The `Position` property value for a subplot is subject to change until the script either refreshes the plot or exits.
- Use `clf` to clear the contents of the figure. For example, you might clear the existing subplot layout from the figure before creating a new subplot layout.

See Also

Functions

[axes](#) | [cla](#) | [clf](#) | [figure](#) | [gca](#)

Properties

[Axes Properties](#)

Introduced before R2006a
