Module 08: Plotting

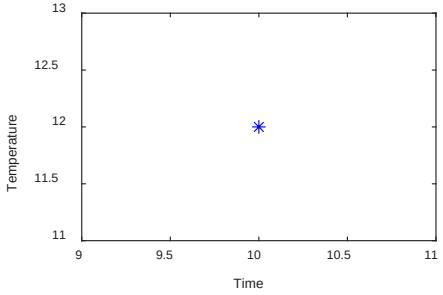
Chul Min Yeum
Assistant Professor
Civil and Environmental Engineering
University of Waterloo, Canada

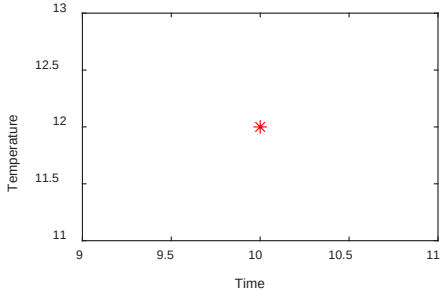
UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

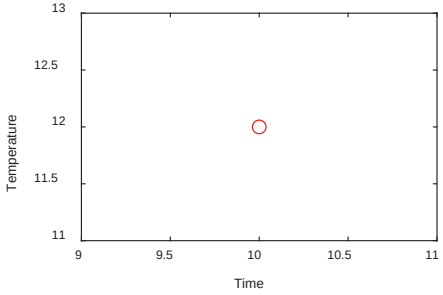
Module 08: Learning Outcomes

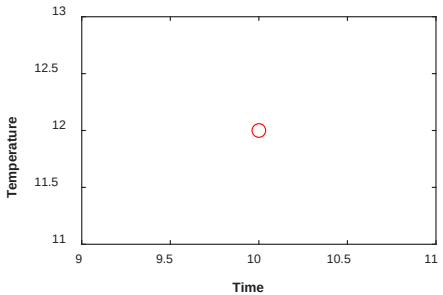
- Create a graph using the '`plot`' function
- Customize your plot: marker color & type, line color, labels, set range, etc
- Draw multiple plots or graphs in a single figure window
- Save your graph

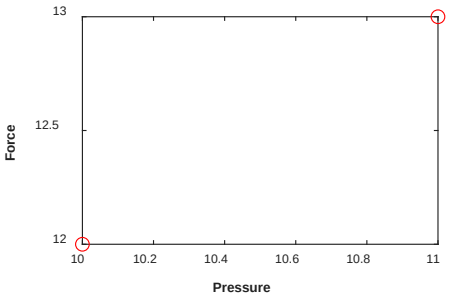
Plot a Point (Overview)

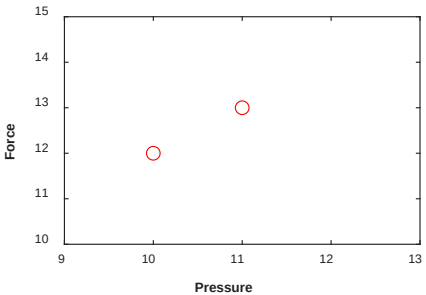






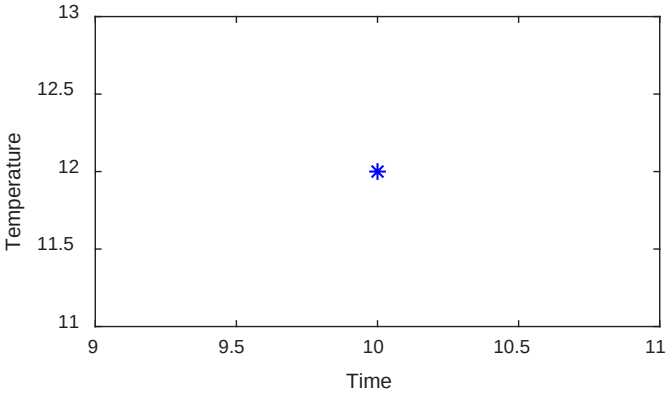




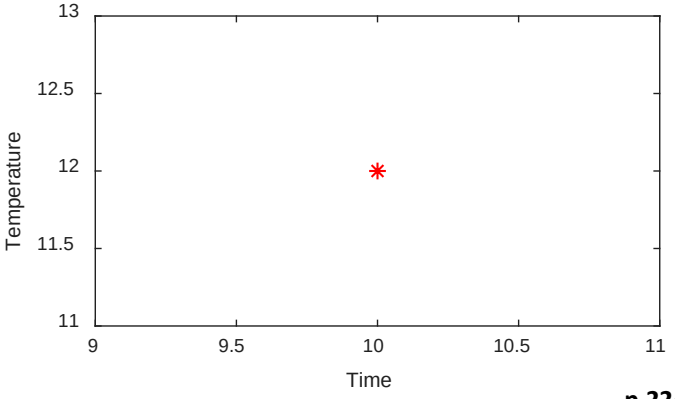


Plot a Point: Change Marker Color

```
x = 10;
y = 12;
plot(x,y, 'b*')

xlabel('Time');
ylabel('Temperature');
```



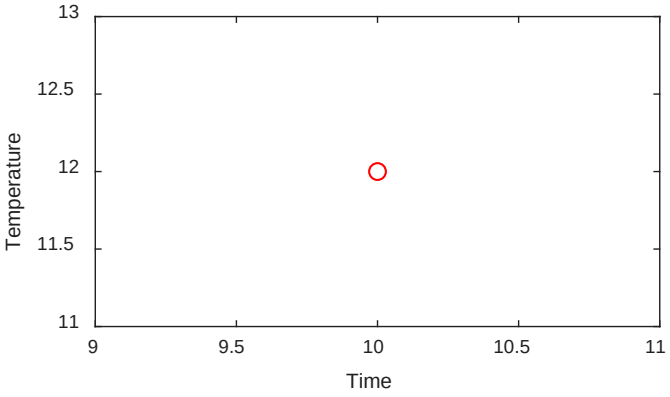

```
x = 10;
y = 12;
plot(x,y, 'r*')

xlabel('Time');
ylabel('Temperature');
```



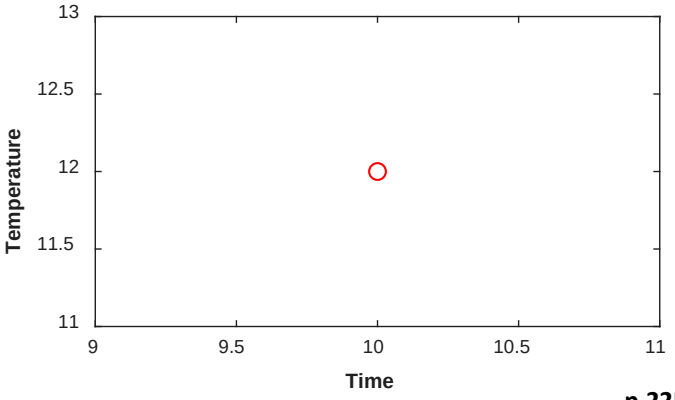
📖: Color can be specified as a character. `'r'`: red, `'b'`: blue, `'m'`: magenta, `'g'`: green, `'k'`: black, `'y'`: yellow.

Plot a Point: Change Marker Type & Make Labels as a Bold Font

```
x = 10;
y = 12;
plot(x,y, 'ro')

xlabel('Time');
ylabel('Temperature');
```




```
x = 10;
y = 12;
plot(x,y, 'ro')

xlabel('\bf Time');
ylabel('\bf Temperature');
```

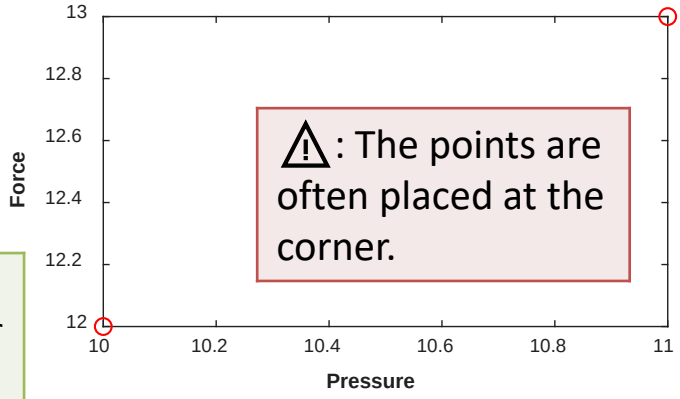


📖: Marker type can be specified as a character. `'o'`: circle, `'x'`: cross, `'*'`: asterisk, `'+'`: plus sign

Plot Point(s): Add Point(s) & Change Min. and Max Value in X & Y-axis

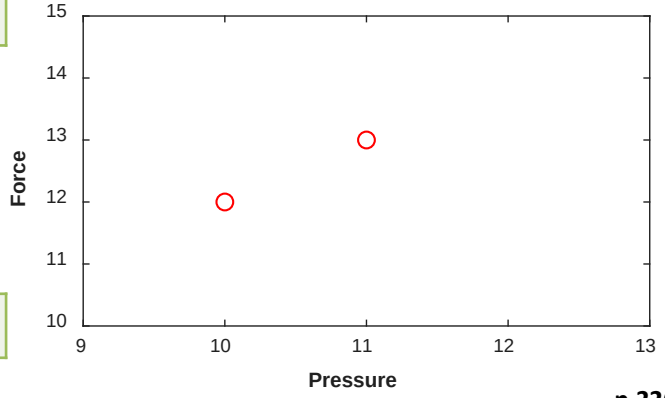
```
x = [10 11];
y = [12 13];
plot(x,y, 'ro')

xlabel('\bf Pressure');
ylabel('\bf Force');
```

📖: x and y are inputted as a vector to plot multiple points.



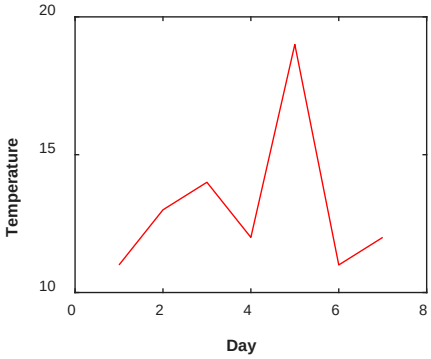
⚠: The points are often placed at the corner.

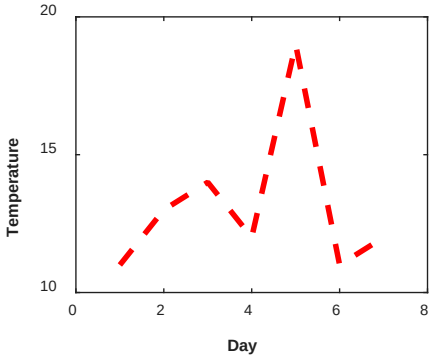
```
x = [10 11];
y = [12 13];
plot(x,y, 'ro')

xlabel('\bf Pressure');
ylabel('\bf Force');
axis([9 13 10 15]);
```

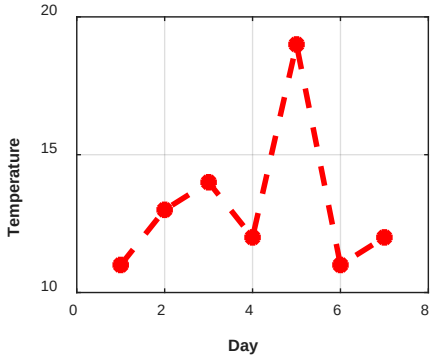


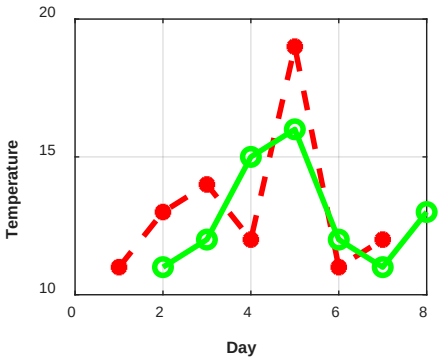
📖: axis([xmin xmax ymin ymax])

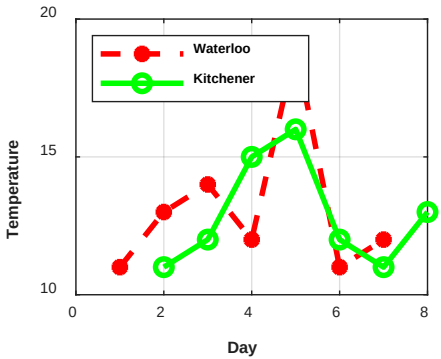
Plot a Vector (Overview)





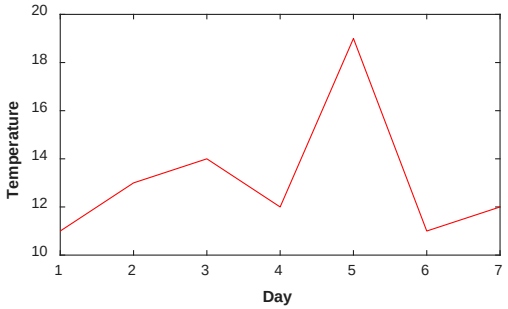






Plot a Vector: Change a Line as a Dashed Line & its Line Width

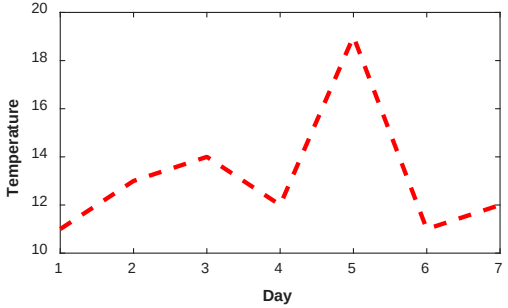
```
x = 1:7;
y = [11 13 14 12 19 11 12];
plot(x,y, 'r')

xlabel('\bf Day');
ylabel('\bf Temperature');
```




```
x = 1:7;
y = [11 13 14 12 19 11 12];
plot(x,y, '--r', 'linewidth', 2)

xlabel('\bf Day');
ylabel('\bf Temperature');
```

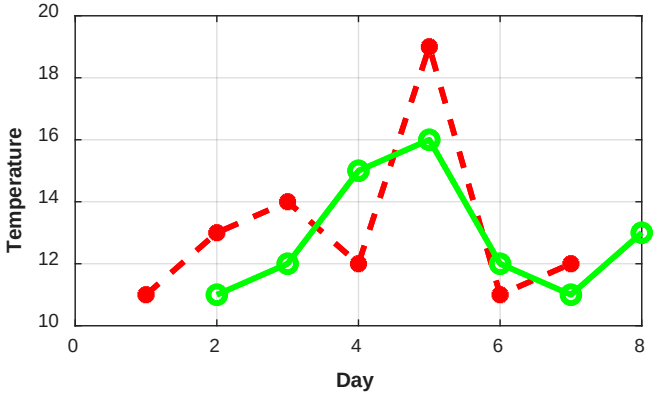
📖: `Plot(… ,'linewidth', value)` is to change line width. As increasing value, the line width is thicker.

📖: If you include color without specifying marker type, the points are connected (default: solid line). Line style character can specify the line style. `'-'`: solid, `'--'`: dash, `':'`: dotted, `'-.'`: dash-dot.



Plot a Vector: Add Data Point Marker and Grid & Add Another Line

```
x = 1:7;
y = [11 13 14 12 19 11 12];
plot(x,y, '--r*', 'linewidth', 2)

hold on;
x2 = 2:8;
y2 = [11 12 15 16 12 11 13];
plot(x2,y2, '-go', 'linewidth', 2)

xlabel('Day'); grid on;
ylabel('Temperature');
```



📖: The way of plotting two graphs in one plot is to overlay (combine) two graphs using `hold on`.

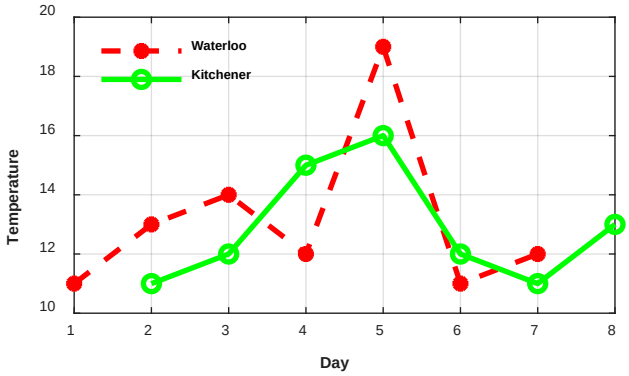
📖: When you combine line and marker style with a color like '`--r*`', you can plot the markers on the line, which are used for drawing the corresponding line.

📖: `grid on` is to display a grid.

😂: By selecting different line styles, color, line width, marker styles, you can draw nice looking graphs!

Plot a Vector: Add Labels & Their Configuration

```
x = 1:7; y = [11 13 14 12 19 11 12];
plot(x,y, '--r*', 'linewidth', 2); hold on

x2 = 2:8;  y2 = [11 12 15 16 12 11 13];
plot(x2,y2, '-go', 'linewidth', 2)

legend('\bf Waterloo', '\bf Kitchener');
legend('location', 'northwest')
legend('boxoff')

xlabel('\bf Day');grid on
ylabel('\bf Temperature');
```



📖: You can create a legend with descriptive labels for each plotted data using `legend(label1, ...)`

📖: You can remove legend background and outline using `legend('boxoff')`.

📖: The graphs might overlap with the legends. Then, you can change the location of the legend using `legend(..., 'Location', lcn)`. Here, `lcn` includes `'north'`, `'south'`, .. `'northwest'`, etc.

Trigonometric Functions

Function	Description	Script	Value
`deg2rad(x)`	Converts degrees to radians	`deg2rad(90)` `deg2rad(180)`	`1.5708` `3.1416`
`rad2deg(x)`	Converts radians to degrees	`rad2deg(pi)` `rad2deg(pi/2)`	`180` `90`

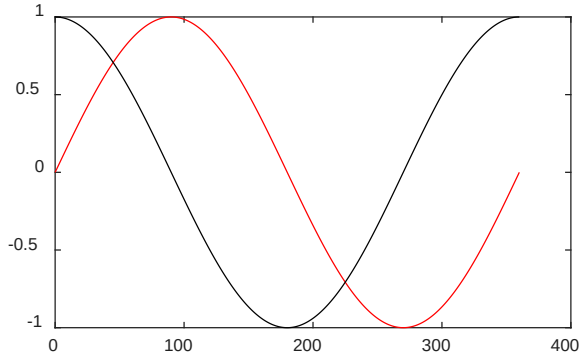
`sin(x)`
`sind(x)`
	Find the sine of **x** when **x** is expressed in **radian/degree**	`sin(pi)`
		`sind(90)`
		`sin(90)`
			`0`
			`1`
			`0.8940`

`cos(x)`
`cosd(x)`
	Find the cosine of **x** when x is expressed in radian/degree	`cos(pi)`
		`cosd(90)`
		`cos(-90)`
			`-1`
			`0`
			`-0.4481`

`asin(x)`
`asind(x)`
	Find the inverse sin of **x** and reports the result in radian (-pi .. pi)/degree(-180 .. 180)	`asin(1)`
		`asind(1)`
		`asind(0.5)`
			`1.5708`
			`90`
			`30`

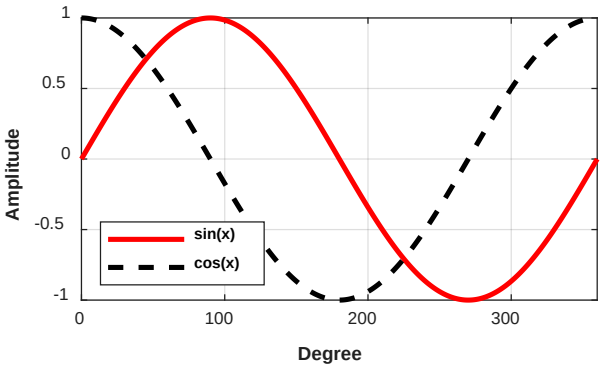
📖: '`-d`' at the end of the name stands for degree.

Example: How to Change and Draw a Graph



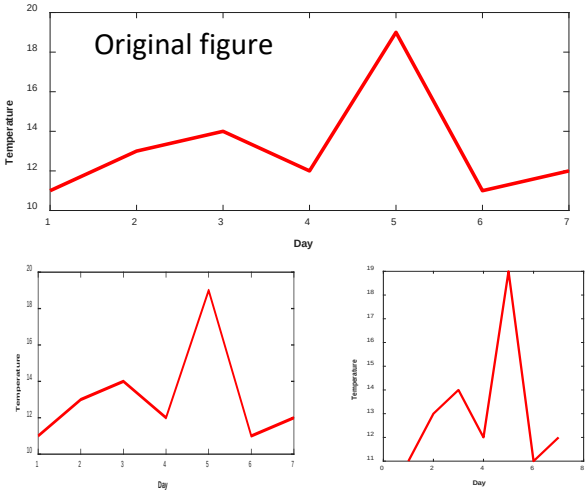

```
x = 0:360;
y1 = sind(x);
y2 = cosd(x);
plot(x, y1,'r'); hold on;
plot(x, y2, ':k')
```

Q. How to change the style of the graph?




```
x = 0:360;
y1 = sind(x);
y2 = cosd(x);
plot(x, y1,'r', 'linewidth', 2); hold
on;
plot(x, y2, '--k', 'linewidth', 2)

legend('\bf sin(x)', '\bf cos(x)');
legend('location', 'southwest')

xlabel('\bf Degree');grid on
ylabel('\bf Amplitude')
axis([0 360 -1 1 ]);
```

Figure Size Control



Resizing with/without keeping aspect ratio of graph contents

```
x = 1:7;
y = [11 13 14 12 19 11 12];

figure(1);
plot(x,y, 'r', 'linewidth', 2)
xlabel('\bf Day');
ylabel('\bf Temperature');
set(gcf, 'Position', [100 100 500 200]);

figure(2);
plot(x,y, 'r', 'linewidth', 2)
xlabel('\bf Day');
ylabel('\bf Temperature');
set(gcf, 'Position', [100 100 300 300]);
```

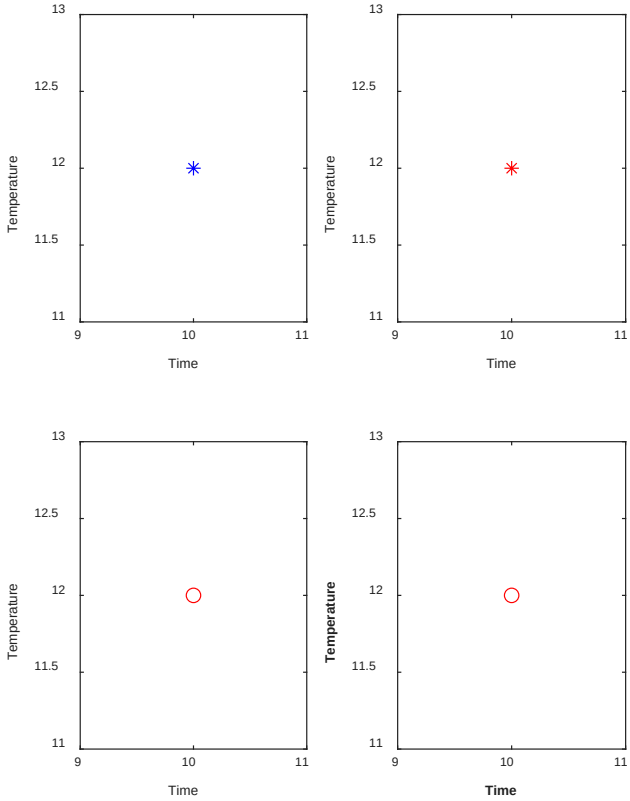
📖: `figure(n)` create a new figure window in which name is n.

📖: When you resize the figure after copying your image, you cannot keep the aspect ratio of graph contents or the original figure. There are two ways to resize figure with the original aspect ratio. Change the figure window before copying the figure or use '`position`' argument.

```
set (gcf, 'position', [left bottom width height])
```

Here, `[left bottom ...]` indicates the location of left-bottom corner of your figure window.

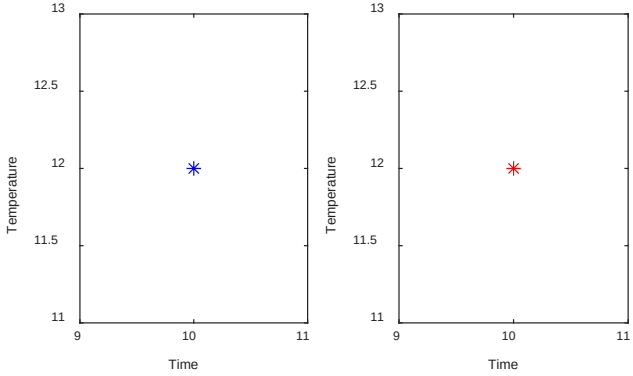
Multiple Figures in One Figure Window

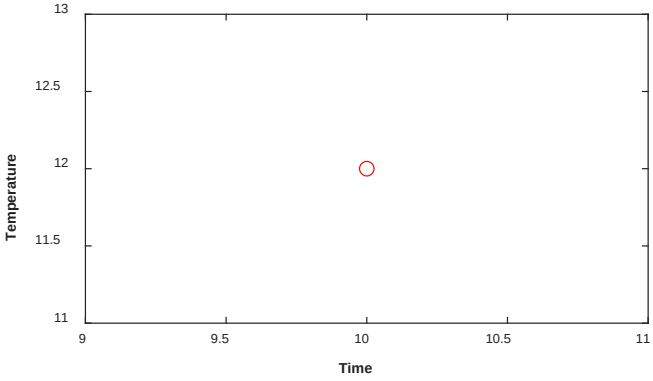



```
figure(1);

x = 10; y = 12;

subplot(2,2,1);
plot(x,y, 'b*')
xlabel('Time'); ylabel('Temperature');

subplot(2,2,2);
plot(x,y, 'r*')
xlabel('Time'); ylabel('Temperature');

subplot(2,2,3);
plot(x,y, 'ro')
xlabel('Time'); ylabel('Temperature');

subplot(2,2,4);
plot(x,y, 'ro')
xlabel('\bf Time');
ylabel('\bf Temperature');

set(gcf,'Position', [100 100 400 500]);
```

Multiple Figures (Continue)

Challenging






```
figure(1);
x = 10; y = 12;

subplot(2,2,1);
plot(x,y, 'b*')
xlabel('Time'); ylabel('Temperature');

subplot(2,2,2);
plot(x,y, 'r*')
xlabel('Time'); ylabel('Temperature');

subplot(2,2,[3 4]);
plot(x,y, 'ro')
xlabel('\bf Time');
ylabel('\bf Temperature');

set(gcf,'Position', [100 100 400 500]);
```

subplot(2,2,n)

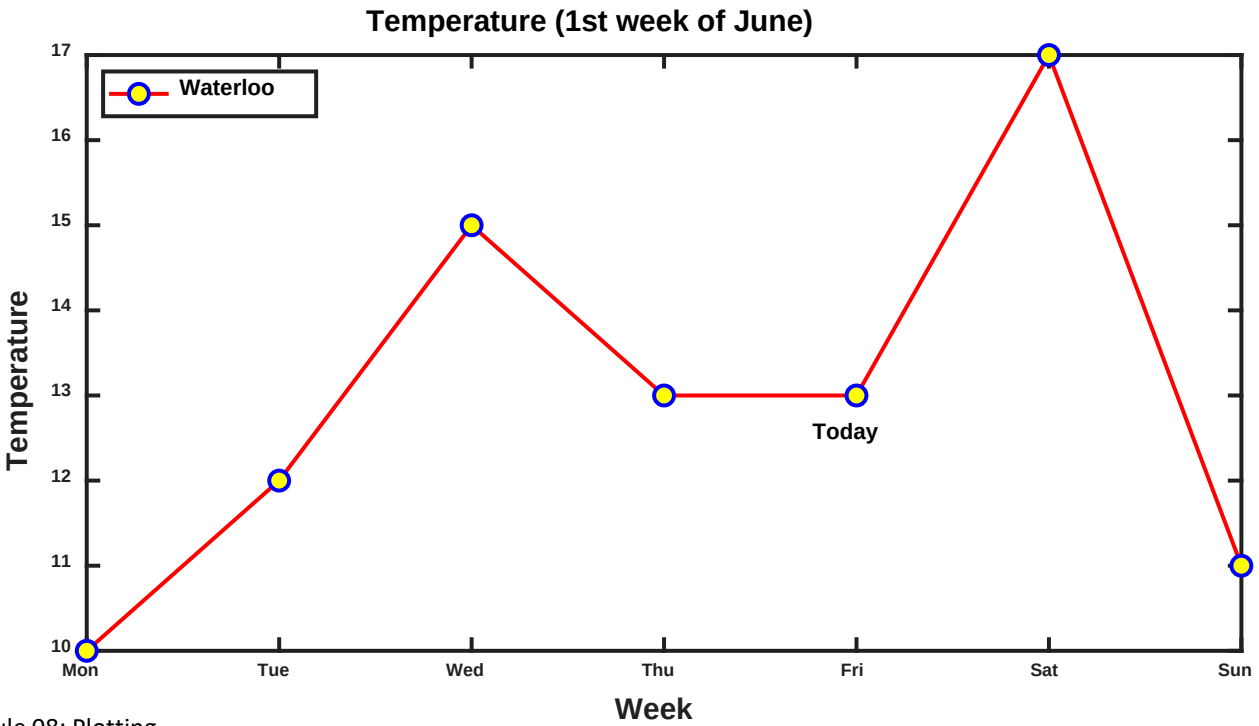
1	2
3	4

subplot(2,3,n)

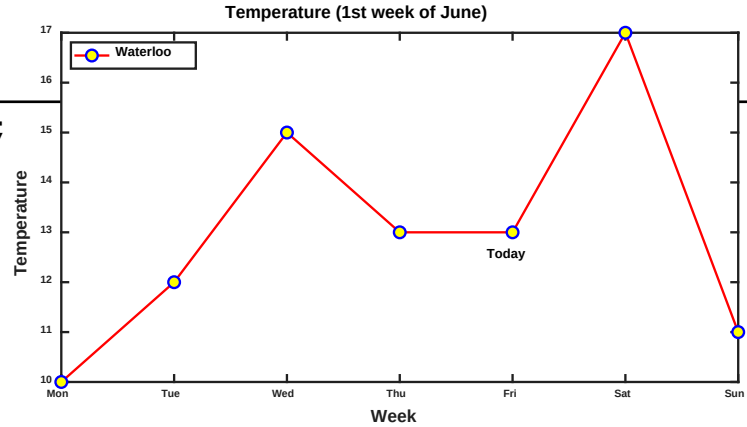
1	2	3
4	5	6

Advanced Plotting

Challenging



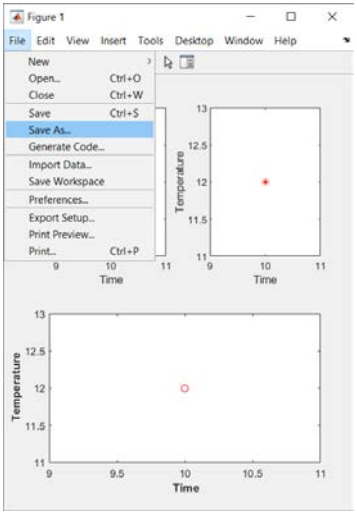
Advanced Plotting (Continue)

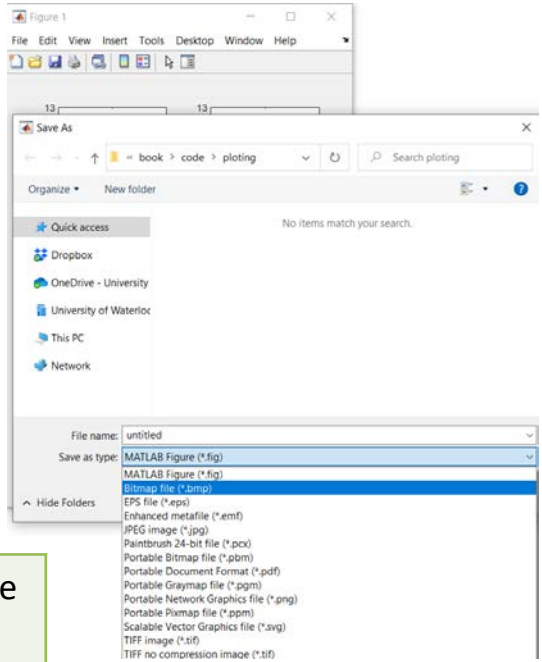


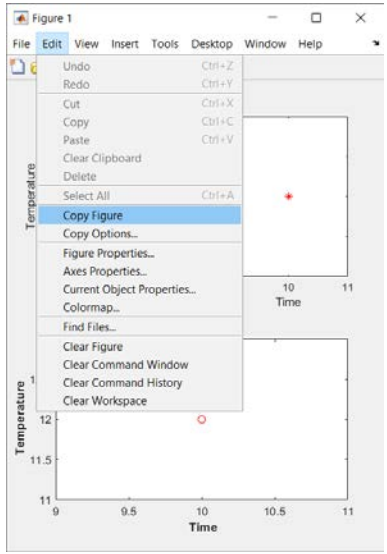

```
x = 1:7; y = [10 12 15 13 13 17 11];
plot(x,y, '-ro', ...
    'LineWidth', 2, ...
    'MarkerSize', 10, ...
    'MarkerEdgeColor', 'b', ...
    'MarkerFaceColor', 'y')

text(x(5), y(5)-0.5, 'Today', ...
    'FontSize' , 15, ...
    'FontWeight', 'bold', ...
    'HorizontalAlignment', 'center');

xticks(1:7);
xticklabels({'Mon','Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'});

legend('\bf Waterloo', 'Location', 'northwest', 'FontSize', 15);
xlabel('\bf Week', 'fontsize', 20);
ylabel('\bf Temperature', 'fontsize', 20);
title('Temperature (1st week of June)', 'fontsize', 20);

set(gca, 'Fontsize', 12, 'FontWeight', 'bold');
set(gca, 'LineWidth', 2);
set(gcf, 'Position', [100 100 1000 500]);
```

📖: `...` is to continue long statements on multiple lines

Save Your Graphs







📖: Save a graph in an image format: Go to "File-Save As" and select file types to save the figure.

📖: You can copy and paste a graph.