Create a simple plot

In [1]:

```
import matplotlib.pyplot as plt
```

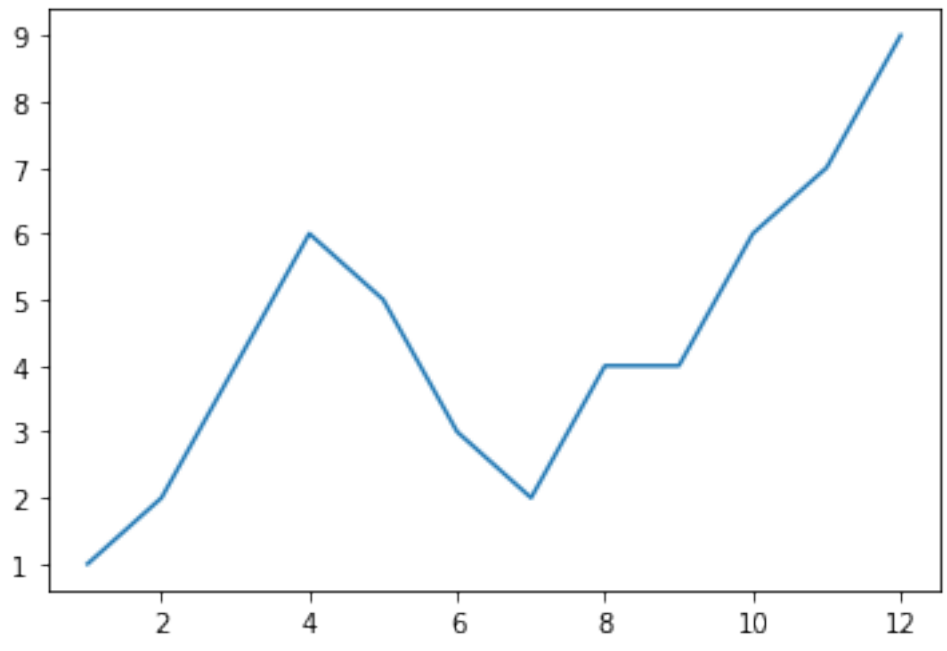
My plot: number of good movies I've seen last year.

x-axis contains integers between 1 and 12 (for each month).
y-axis contains movie numbers

Plot() function documentation: http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot (http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [2]:

```
# magic function that renders the figure in a notebook
%matplotlib inline

# y-axis values
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]

# define my plot: x-axis is from 1 to 12
plt.plot(range(1, 13), movies_per_month )


#display
plt.show()
```



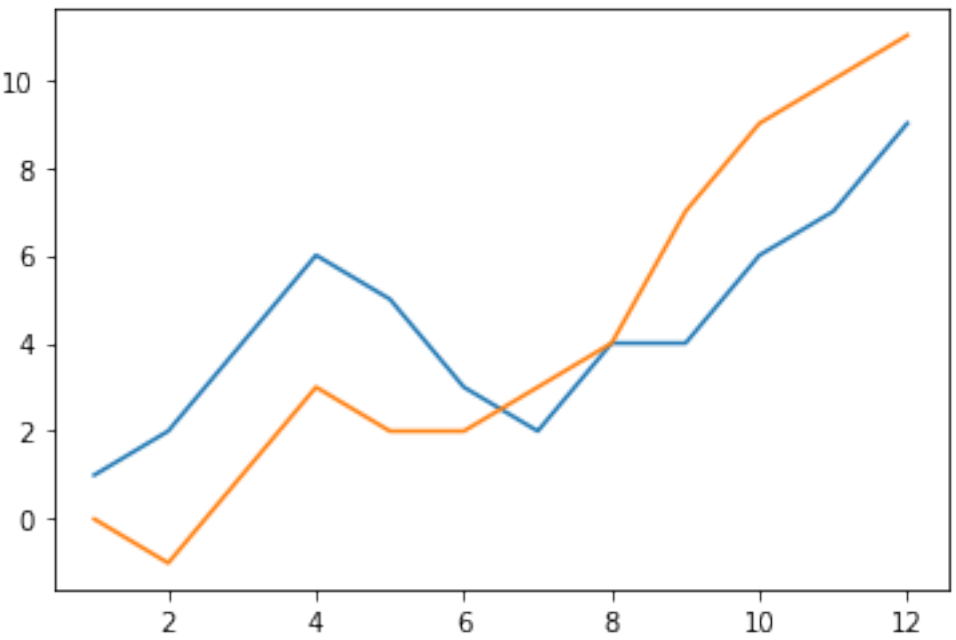
Drawing multiple lines and plots.

Display number of good movies and good jokes per month.

To display two plot lines in one graph, simply call plt.plot() multiple times - once for each plot line.
Color of each graph is different so that you can tell them apart.

In [3]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

# Create two plots
plt.plot(range(1, 13), movies_per_month)
plt.plot(range(1, 13), jokes_per_month)

#display
plt.show()
```



Creating a legend

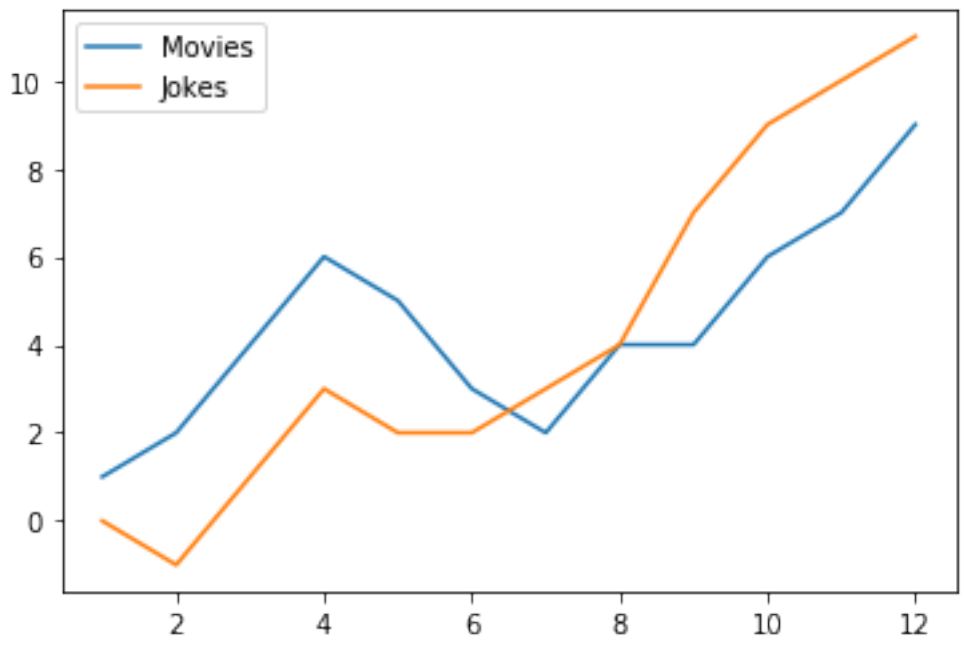
Legend location codes:

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.legend

In [4]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

line1 = plt.plot(range(1,13), movies_per_month)
line2 = plt.plot(range(1,13), jokes_per_month)

# Add a legend
plt.legend(['Movies', 'Jokes'], loc=0)

plt.show()
```



Formatting the axes

The axes define the x and y plane of the graphic. The x axis runs horizontally, and the y axis runs vertically.

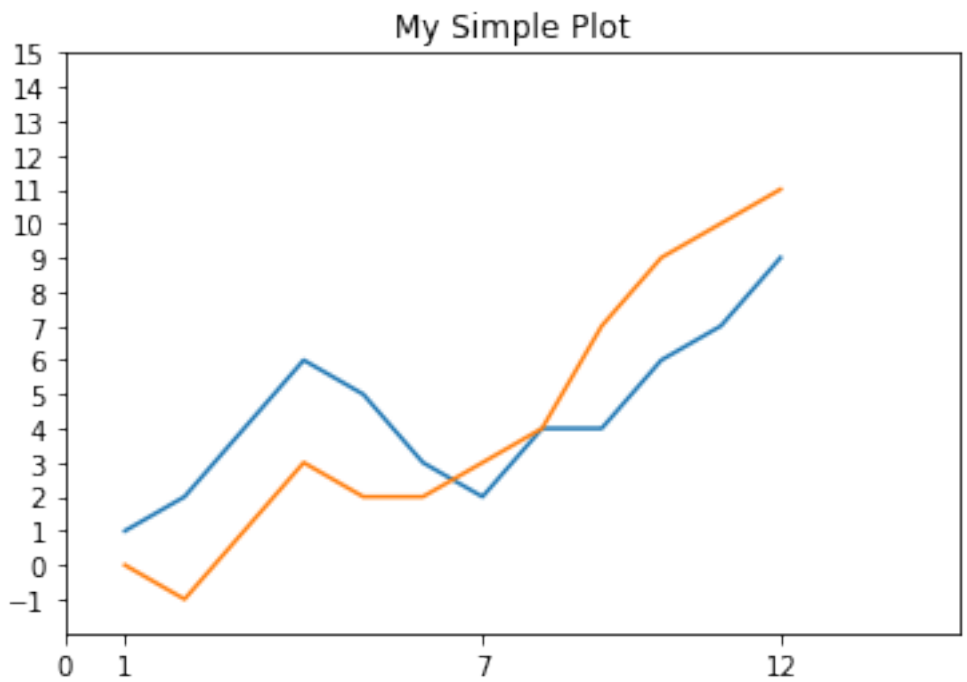
Set the data limits for the x and y axis:
http://matplotlib.org/devdocs/api/_as_gen/matplotlib.axes.Axes.set_xlim.html
(http://matplotlib.org/devdocs/api/_as_gen/matplotlib.axes.Axes.set_xlim.html)

Set the ticks with list of ticks.

Axes documentation: http://matplotlib.org/api/axes_api.html (http://matplotlib.org/api/axes_api.html)

In [14]:

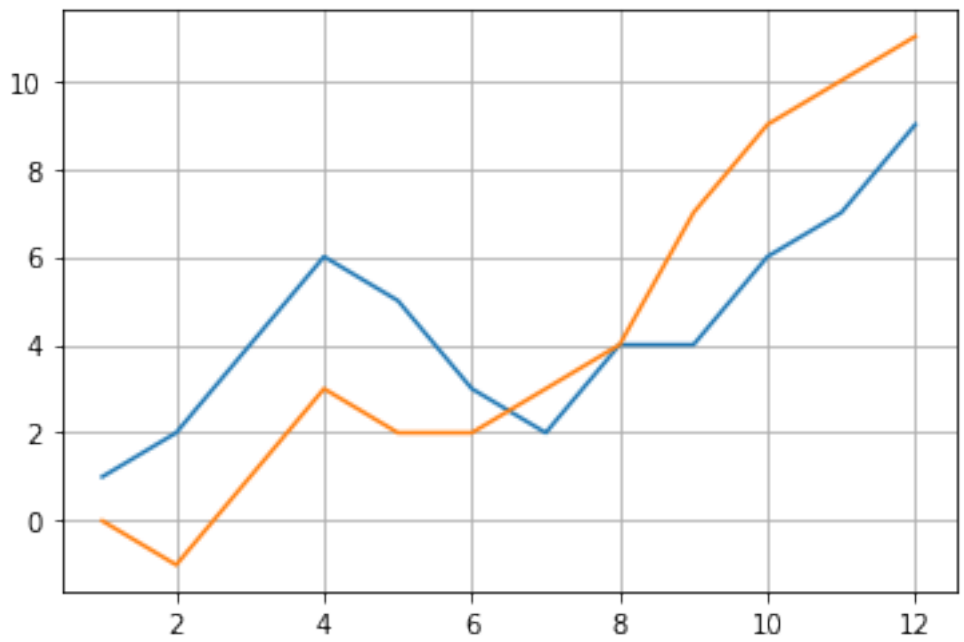
```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

# define both plots
line1 = plt.plot(range(1,13), movies_per_month)
line2 = plt.plot(range(1,13), jokes_per_month)

# get the current axes
ax = plt.gca()

ax.set_title("My Simple Plot")

# Set the data limits for the x and y axis
ax.set_xlim([0, 15])
ax.set_ylim([-2, 15])

# Set the ticks
ax.set_xticks([0, 1, 7, 12])
ax.set_yticks([-1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])

plt.show()
```



Adding grids

In [6]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

plt.plot(range(1,13), movies_per_month)
plt.plot(range(1,13), jokes_per_month)

# get the current axes and add grids
ax = plt.gca()
ax.grid(True)

plt.show()
```



Defining the Line Appearance

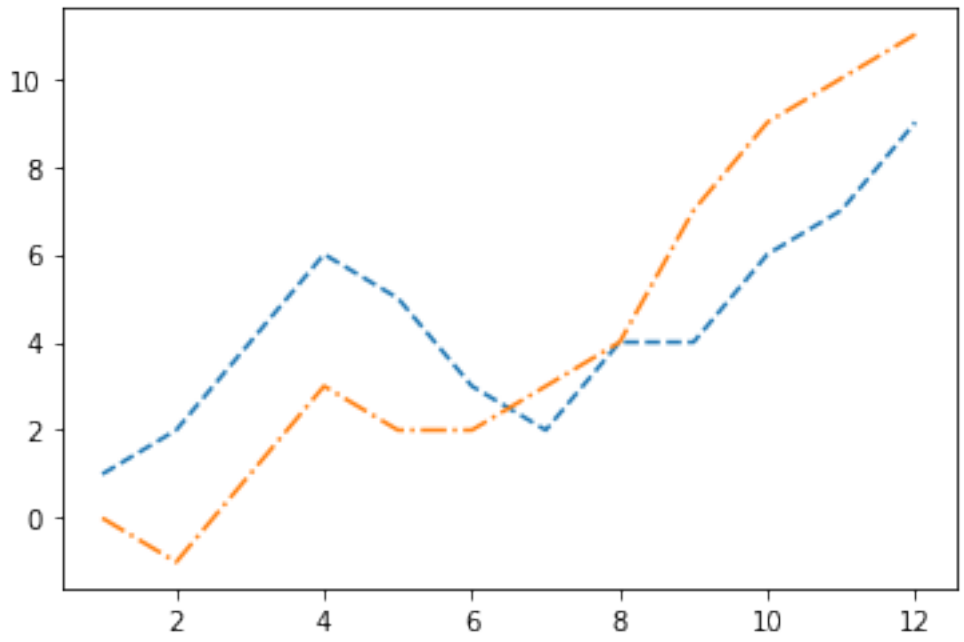
Choosing a line style

Characters	Line Style
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

For a complete list of line styles, see plot function documentation:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [16]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

# Create two plots
plt.plot(range(1, 13), movies_per_month, '--') # dashed line
plt.plot(range(1, 13), jokes_per_month, linestyle='-.') # dash-dot line style

#display
plt.show()
```



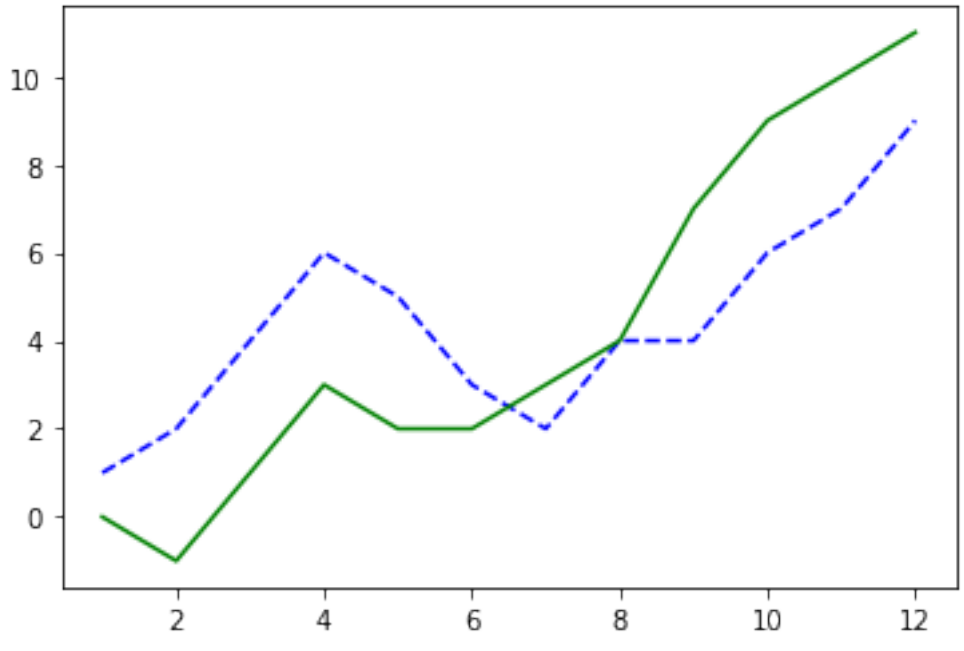
Choosing a line color

Characters	Line Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'k'	black
'w'	white

See plot function documentation for a supported list of line colors:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [15]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

# red and blue
plt.plot(range(1, 13), movies_per_month, 'b--')
plt.plot(range(1, 13), jokes_per_month, color='g')

plt.show()
```



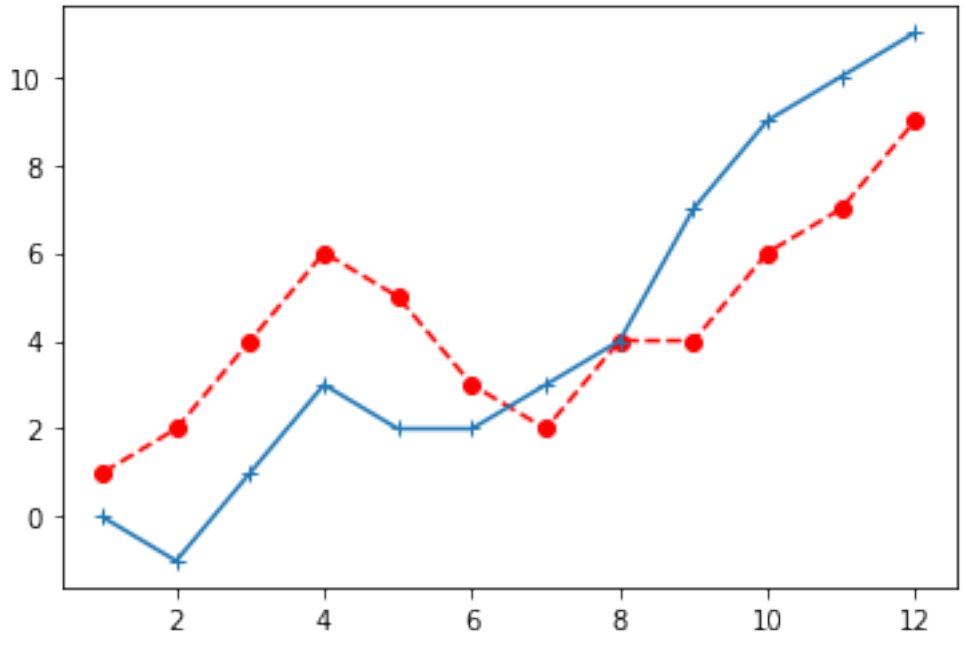
Choosing a marker

Characters	Marker
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

See plot function documented for a list of supported markers:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [18]:

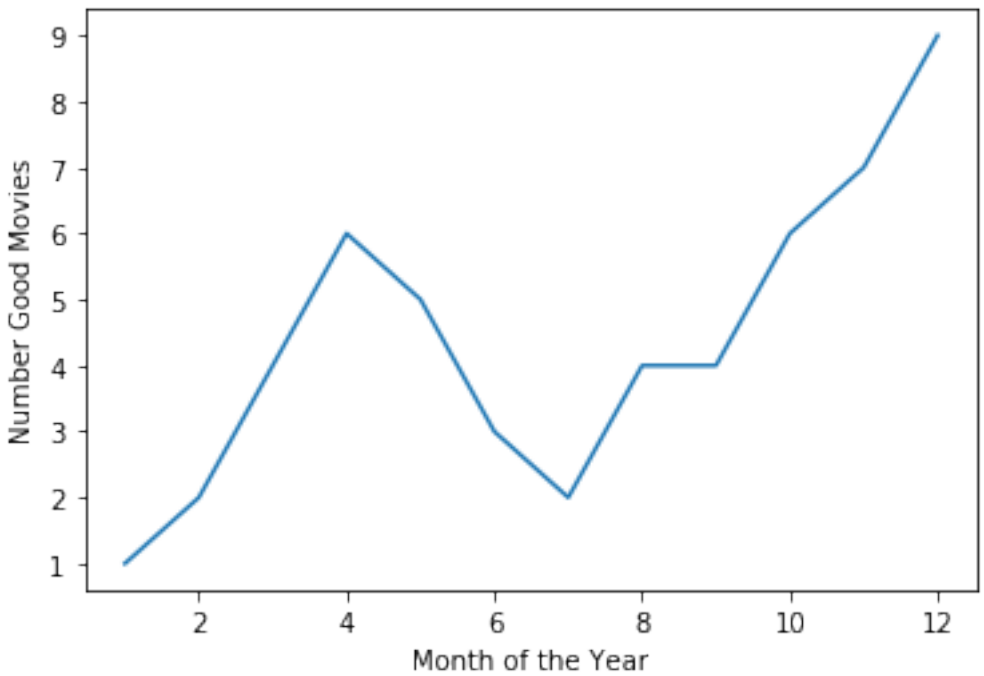
```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]
jokes_per_month = [0, -1, 1, 3, 2, 2, 3, 4, 7, 9, 10, 11]

# markers
plt.plot(range(1, 13), movies_per_month, 'ro--') # red circle marker and dashed line
plt.plot(range(1, 13), jokes_per_month, marker='+') # plus marker

plt.show()
```



Adding labels for x and y axis

In [10]:

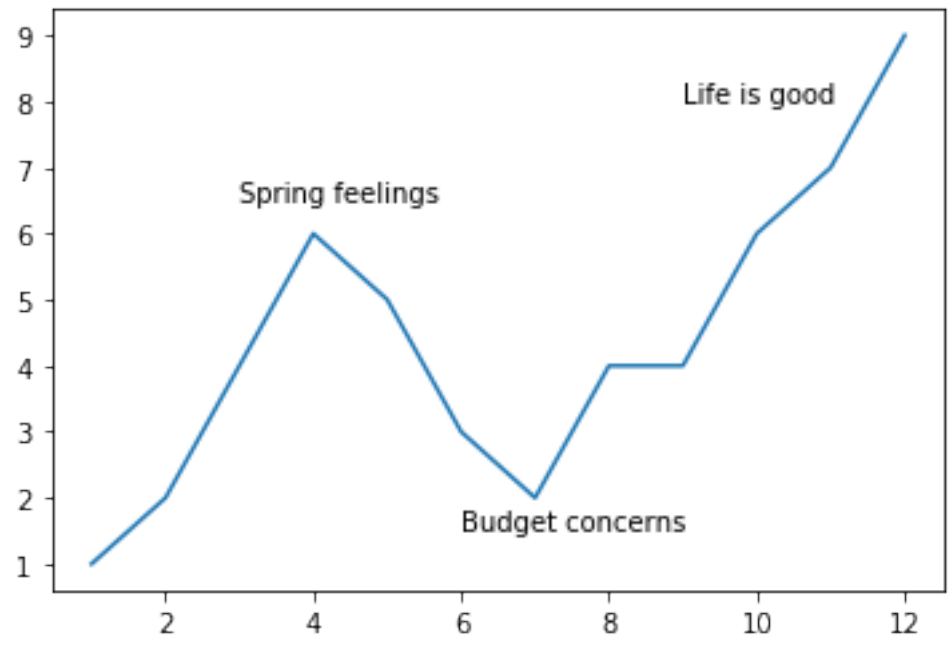
```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]

# labels
plt.xlabel('Month of the Year')
plt.ylabel('Number Good Movies')

plt.plot(range(1, 13), movies_per_month)

plt.show()
```



Annotating the chart

In [11]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]

# add annotation
plt.annotate(xy=[3, 6.5], s='Spring feelings')
plt.annotate(xy=[6, 1.5], s='Budget concerns')
plt.annotate(xy=[9, 8], s='Life is good')

plt.plot(range(1, 13), movies_per_month)

plt.show()
```

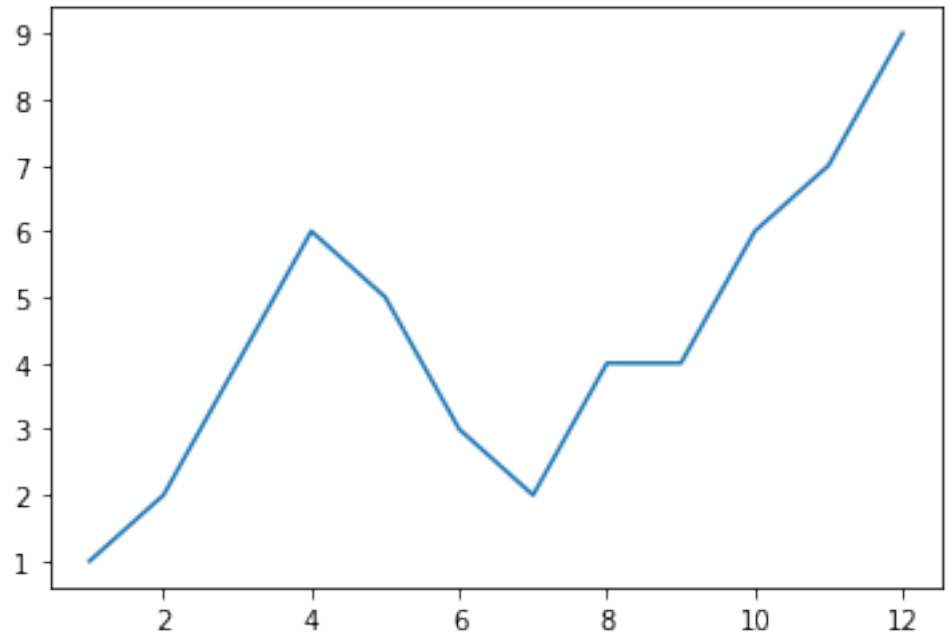


Save the graph

Save the current figure with **plt.savefig(filename, format):**
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig
[(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig)](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig)

In [12]:

```
movies_per_month = [1, 2, 4, 6, 5, 3, 2, 4, 4, 6, 7, 9]

plt.plot(range(1, 13), movies_per_month)

# save the current figure
plt.savefig('MySamplePlot.png', format='png')
```



Get the Available Supported File Format

In [13]:

```
fig = plt.figure()

print (fig.canvas.get_supported_filetypes())
```

```
{'ps': 'Postscript', 'eps': 'Encapsulated Postscript', 'pdf': 'Portabl
e Document Format', 'pgf': 'PGF code for LaTeX', 'png': 'Portable Netw
ork Graphics', 'raw': 'Raw RGBA bitmap', 'rgba': 'Raw RGBA bitmap', 's
vg': 'Scalable Vector Graphics', 'svgz': 'Scalable Vector Graphics', '
jpg': 'Joint Photographic Experts Group', 'jpeg': 'Joint Photographic
Experts Group', 'tif': 'Tagged Image File Format', 'tiff': 'Tagged Ima
ge File Format'}

<Figure size 432x288 with 0 Axes>
```

In [ ]:

Create a simple plot

In [1]:

```
```

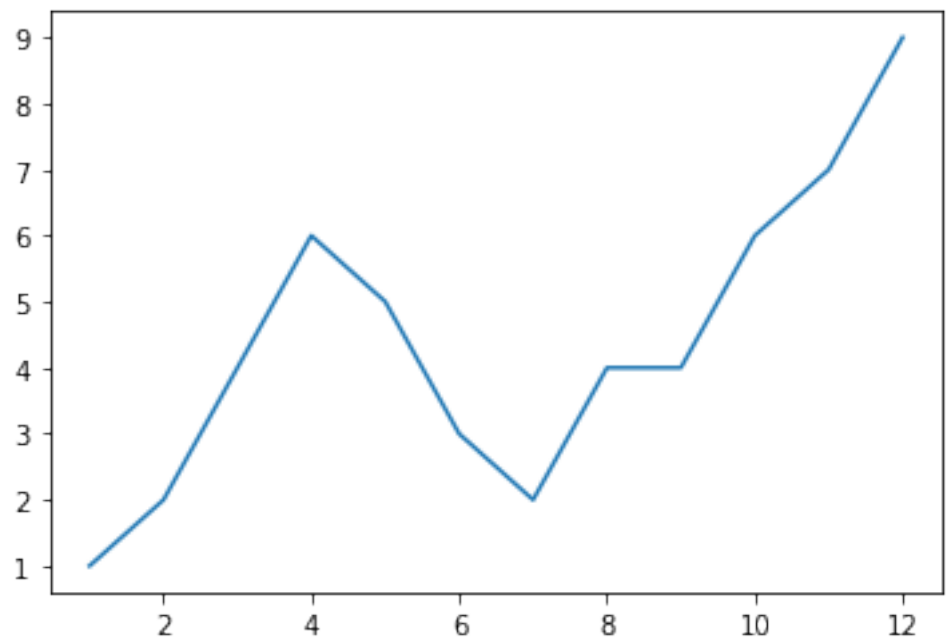
My plot: number of good movies I've seen last year.

x-axis contains integers between 1 and 12 (for each month).
y-axis contains movie numbers

Plot() function documentation: http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot (http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [2]:

```
```

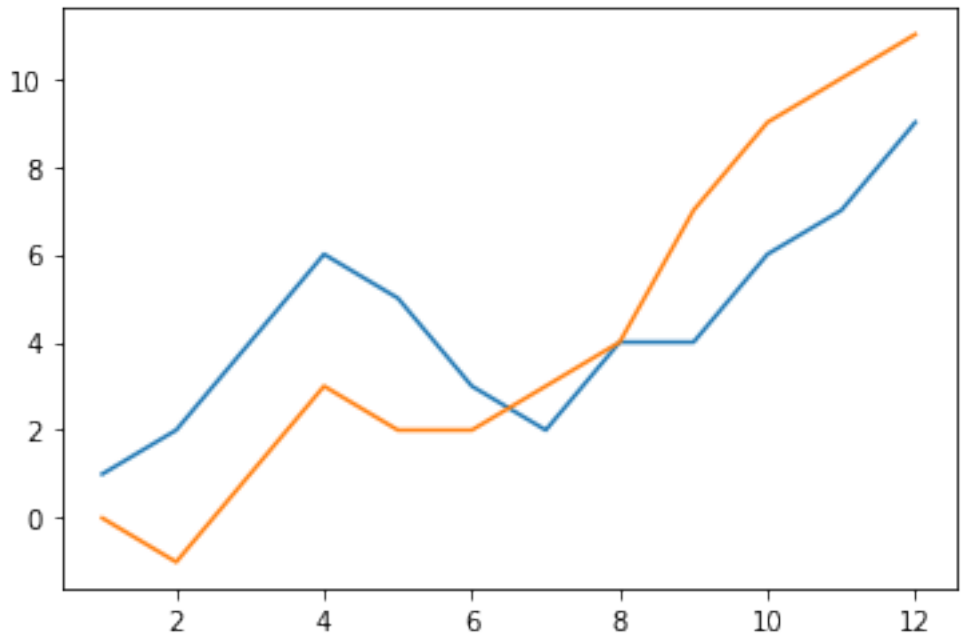


Drawing multiple lines and plots.

Display number of good movies and good jokes per month.

To display two plot lines in one graph, simply call plt.plot() multiple times - once for each plot line.
Color of each graph is different so that you can tell them apart.

In [3]:



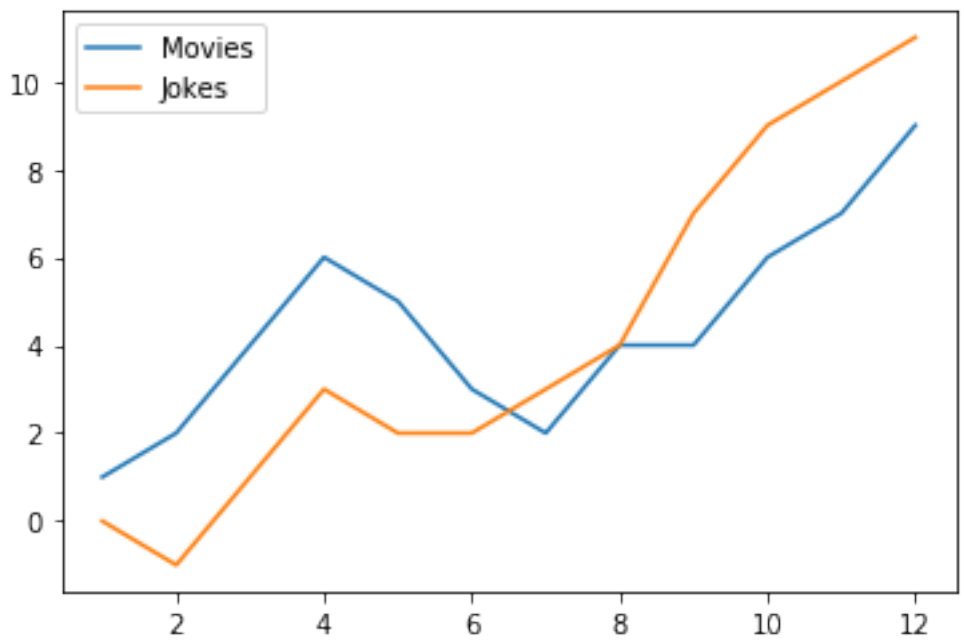
Creating a legend

Legend location codes:

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.legend

In [4]:



Formatting the axes

The axes define the x and y plane of the graphic. The x axis runs horizontally, and the y axis runs vertically.

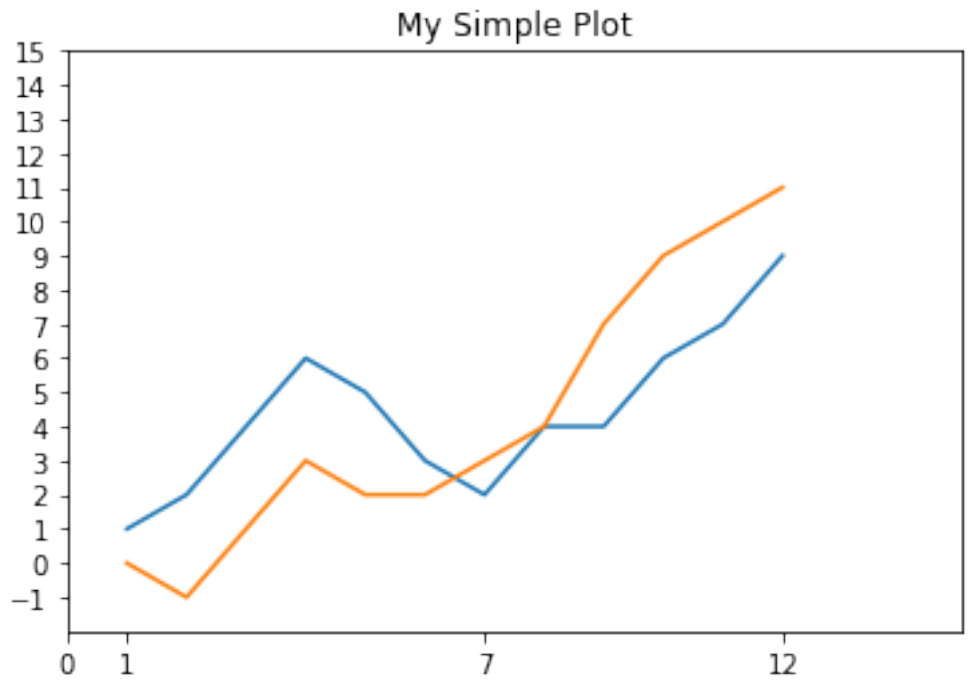
Set the data limits for the x and y axis:

http://matplotlib.org/devdocs/api/_as_gen/matplotlib.axes.Axes.set_xlim.html
(http://matplotlib.org/devdocs/api/_as_gen/matplotlib.axes.Axes.set_xlim.html)

Set the ticks with list of ticks.

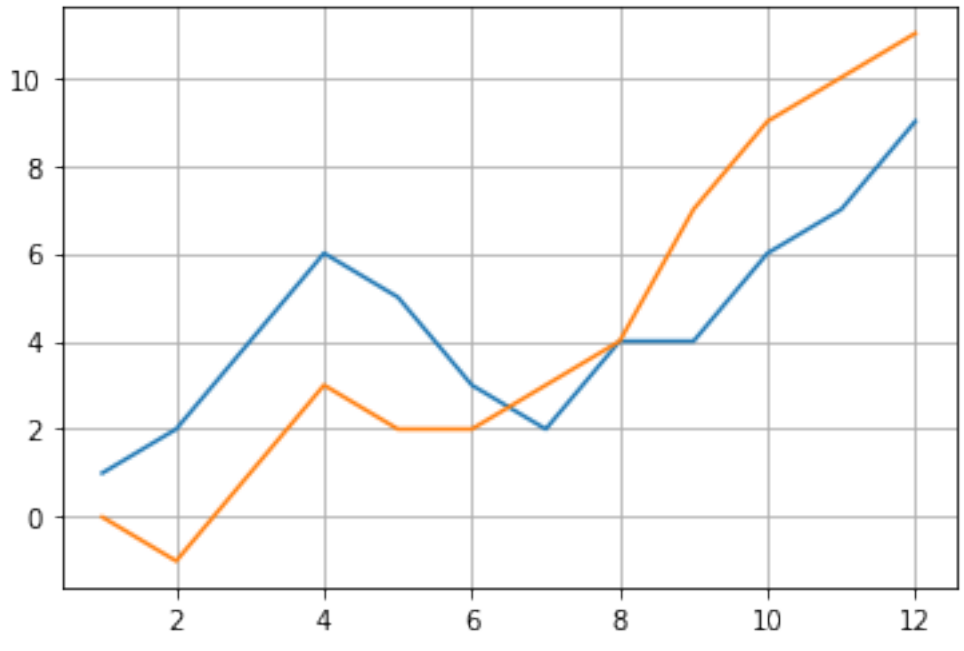
Axes documentation: http://matplotlib.org/api/axes_api.html (http://matplotlib.org/api/axes_api.html)

In [14]:



Adding grids

In [6]:



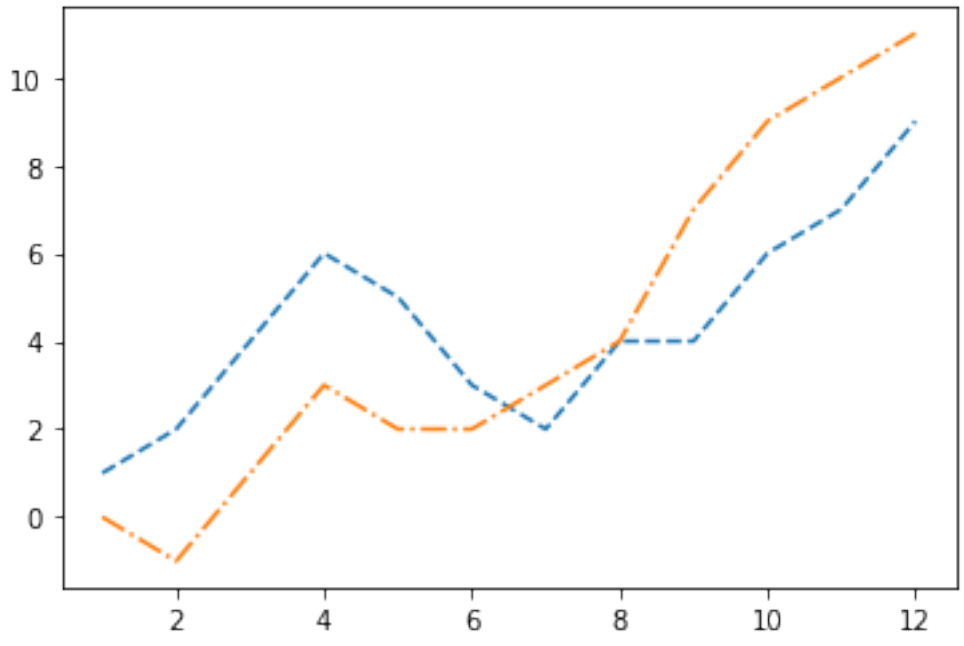
Defining the Line Appearance

Choosing a line style

Characters	Line Style
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style

For a complete list of line styles, see plot function documentation:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [16]:

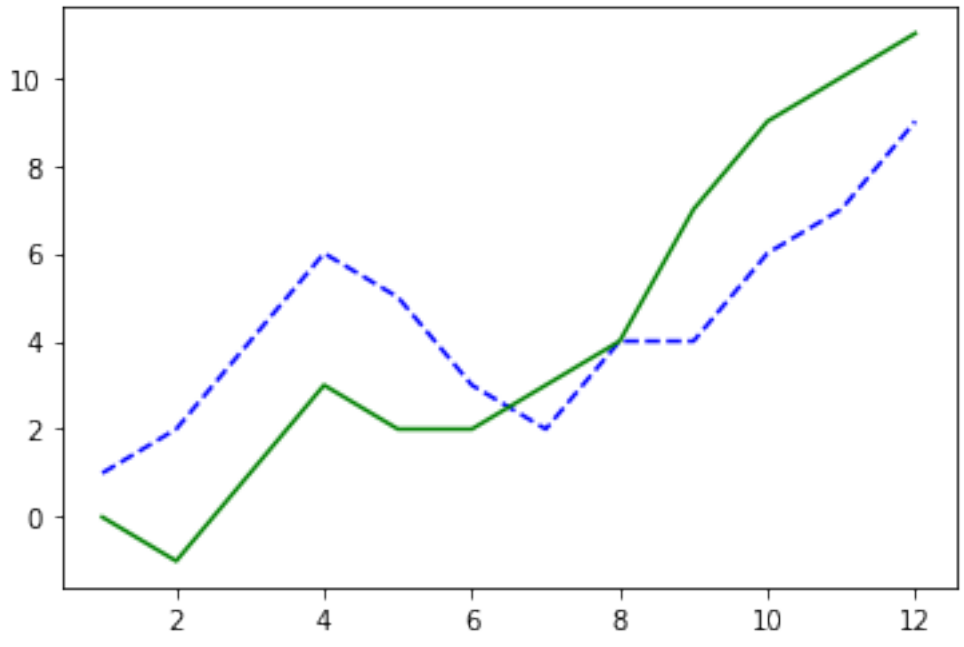


Choosing a line color

Characters	Line Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'k'	black
'w'	white

See plot function documentation for a supported list of line colors:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [15]:

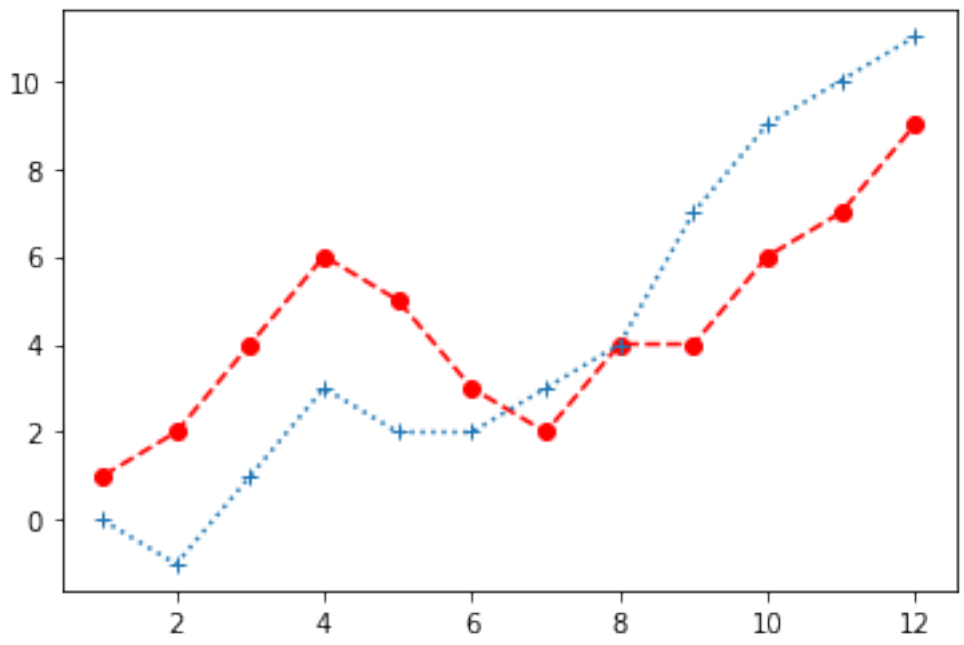


Choosing a marker

Characters	Marker
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

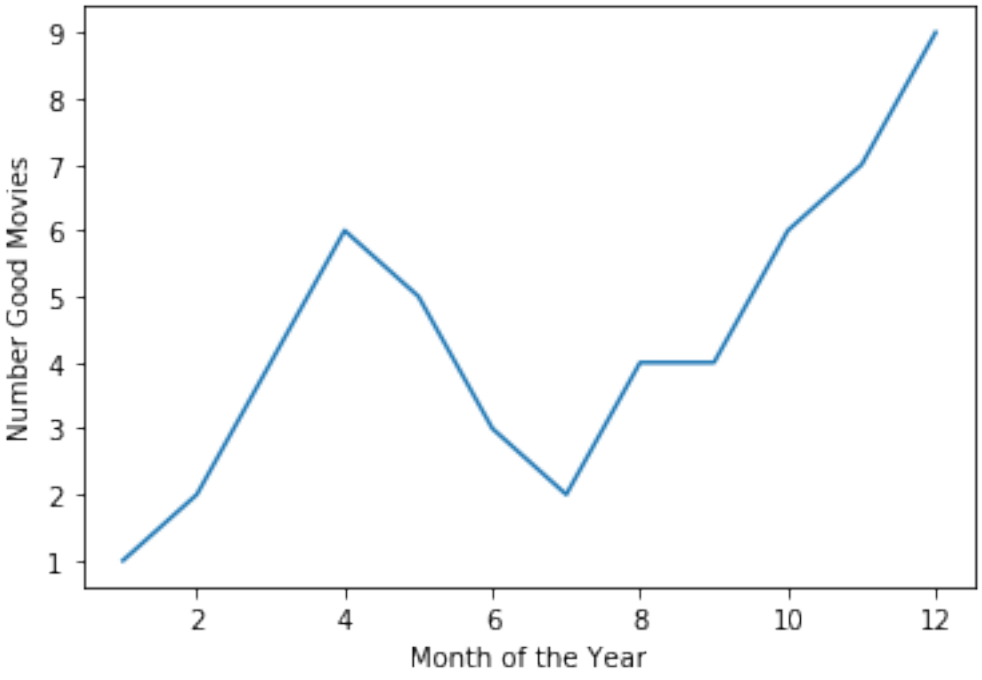
See plot function documented for a list of supported markers:
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot
(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.plot)

In [9]:



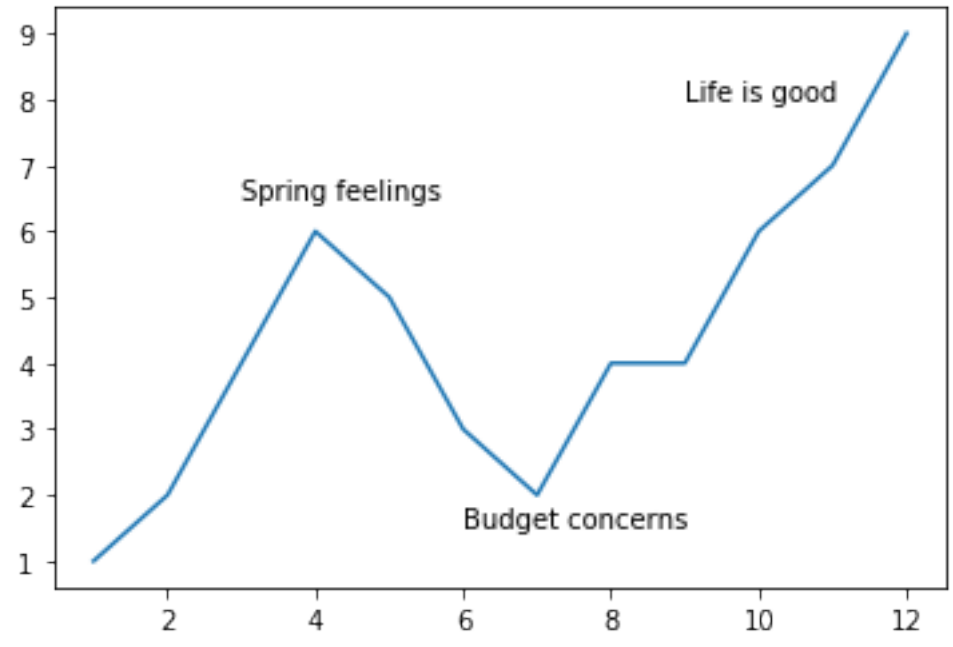
Adding labels for x and y axis

In [10]:



Annotating the chart

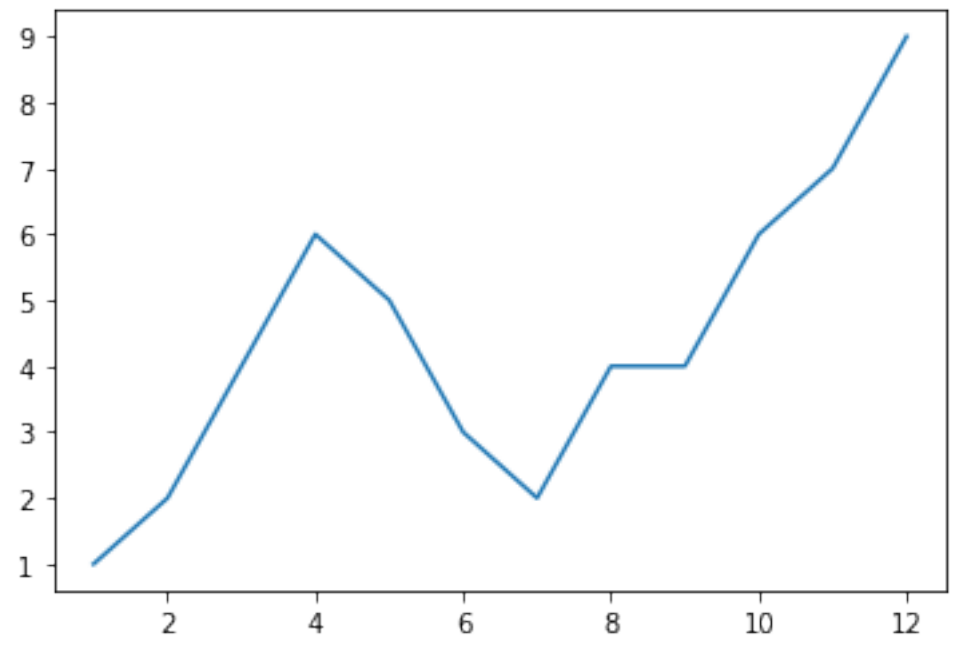
In [11]:



Save the graph

Save the current figure with **plt.savefig(filename, format):**
http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig
[(http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig)](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.savefig)

In [12]:



Get the Available Supported File Format

In [13]:

```
{'ps': 'Postscript', 'eps': 'Encapsulated Postscript', 'pdf': 'Portabl
e Document Format', 'pgf': 'PGF code for LaTeX', 'png': 'Portable Netw
ork Graphics', 'raw': 'Raw RGBA bitmap', 'rgba': 'Raw RGBA bitmap', 's
vg': 'Scalable Vector Graphics', 'svgz': 'Scalable Vector Graphics', '
jpg': 'Joint Photographic Experts Group', 'jpeg': 'Joint Photographic
Experts Group', 'tif': 'Tagged Image File Format', 'tiff': 'Tagged Ima
ge File Format'}

<Figure size 432x288 with 0 Axes>
```

In [ ]: