ISE 314X
Computer Programing for Engineers

MatPlotLib and Data Visualization

Yong Wang
Assistant Professor
Systems Science & Industrial Engineering
Binghamton University

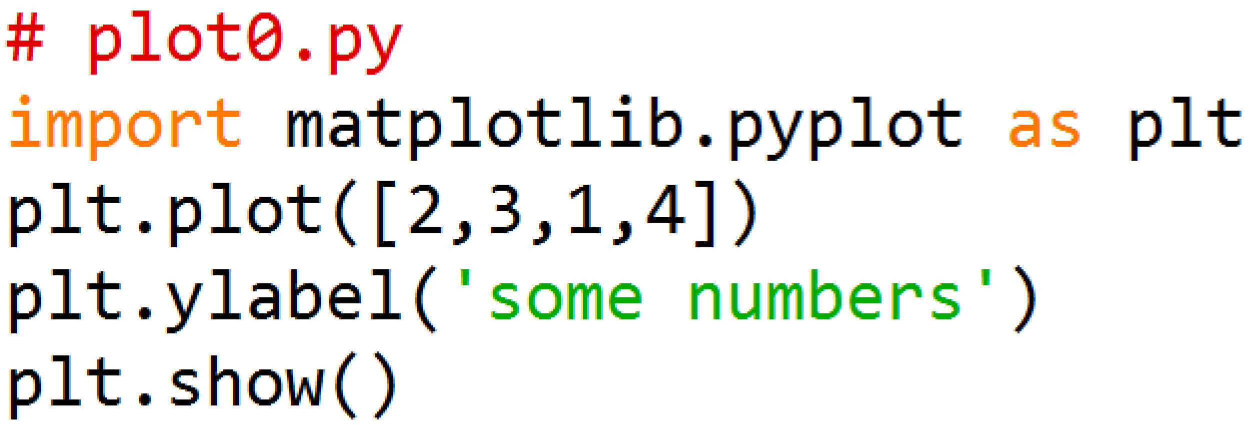
Objectives

- To learn to create high quality 2D and 3D figures from data

The Basics

- matplotlib is a library for making plots of arrays
- matplotlib.pyplot is a collection of functions that make changes to a figure

First Steps



```python
# plot0.py
import matplotlib.pyplot as plt
plt.plot([2,3,1,4])
plt.ylabel('some numbers')
plt.show()
```

First Steps

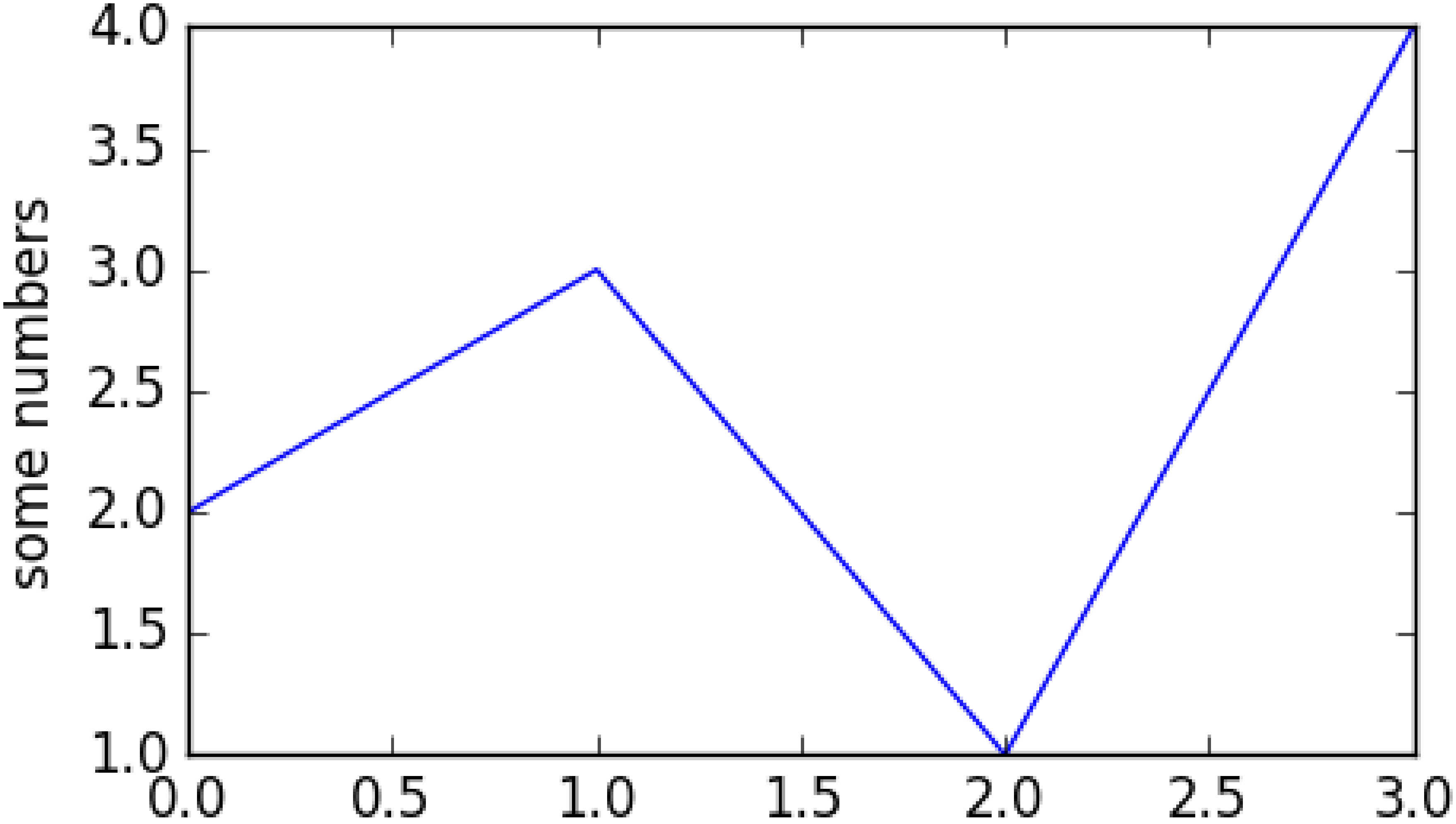
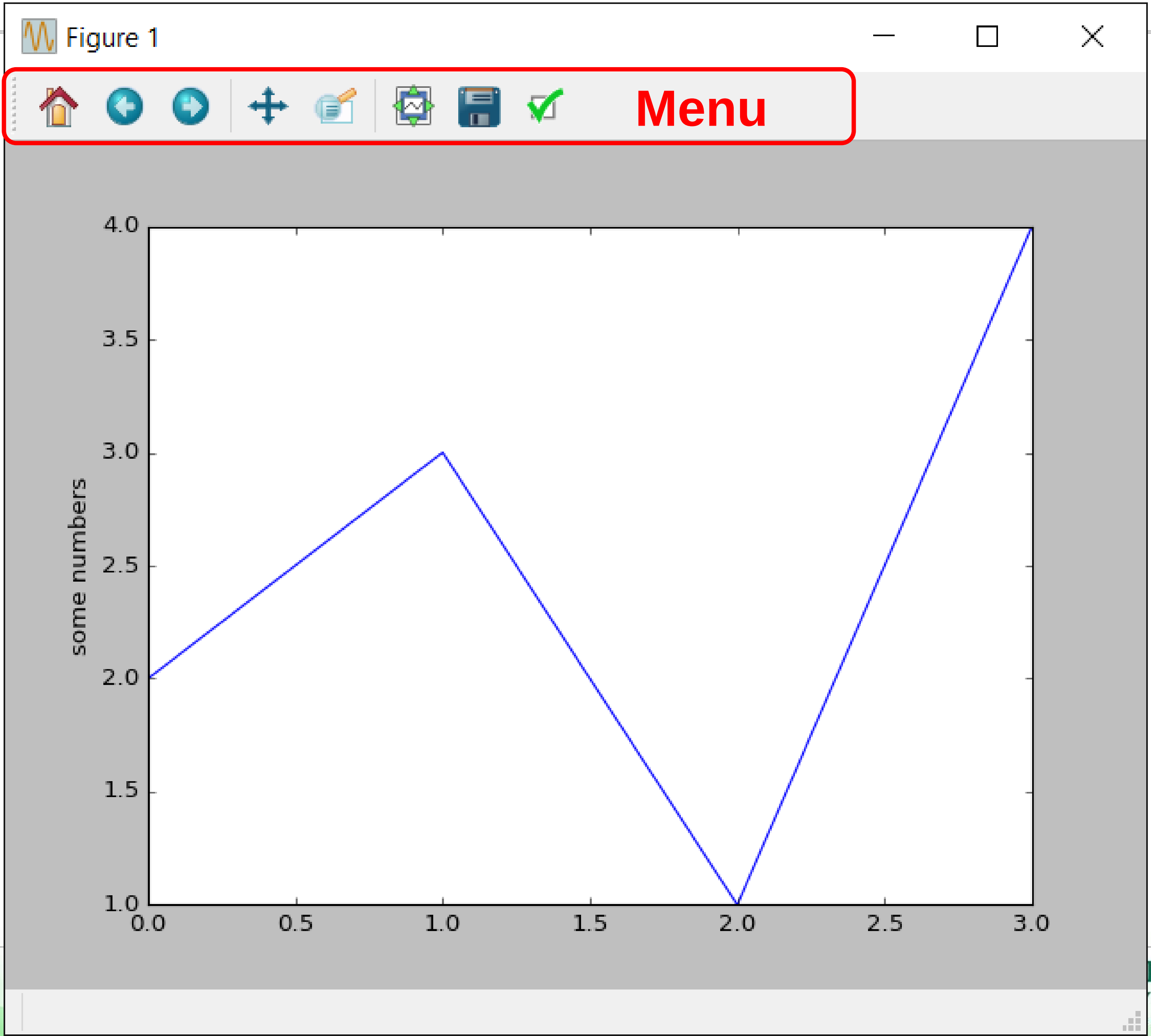
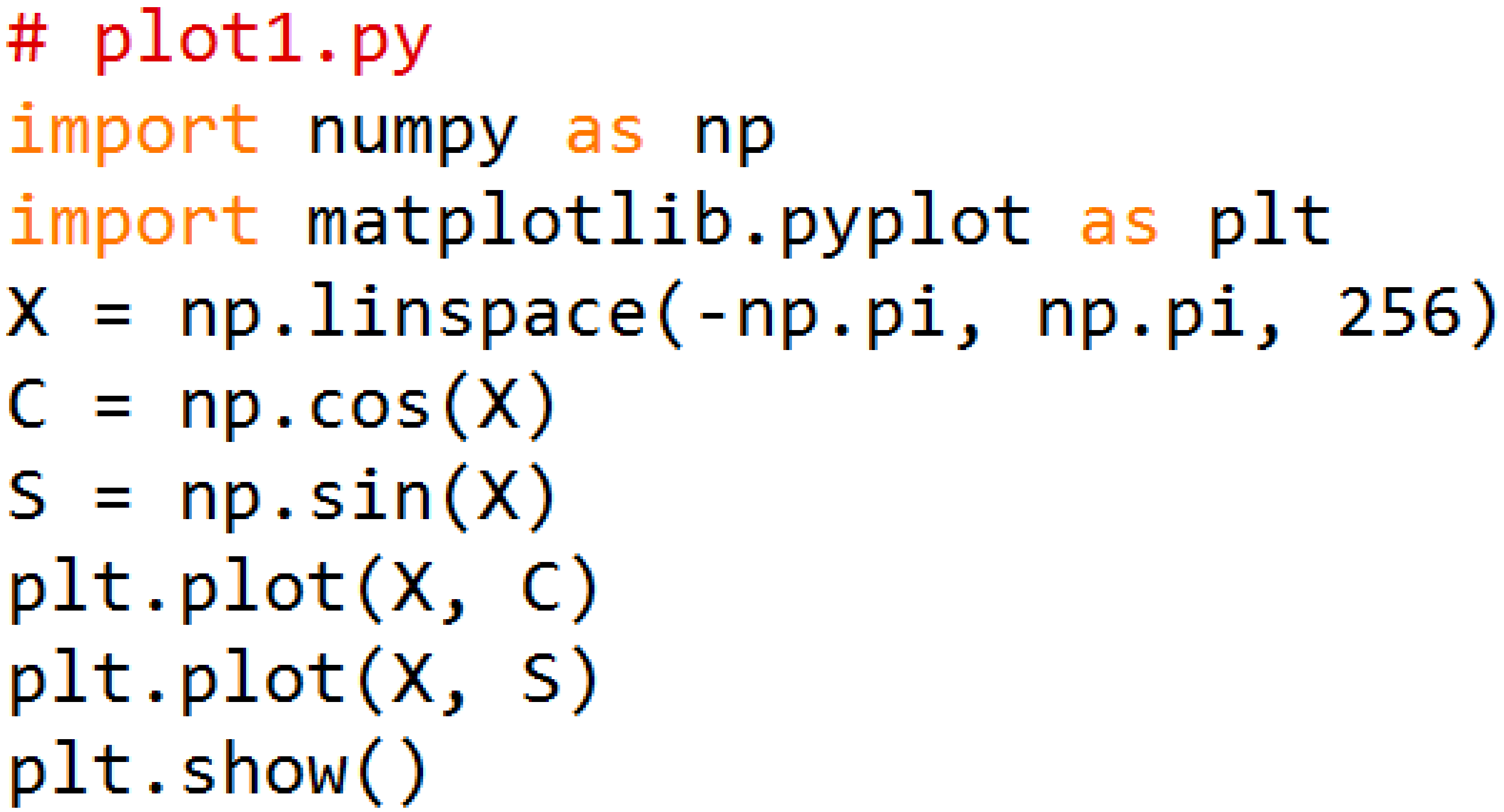


Figure Window

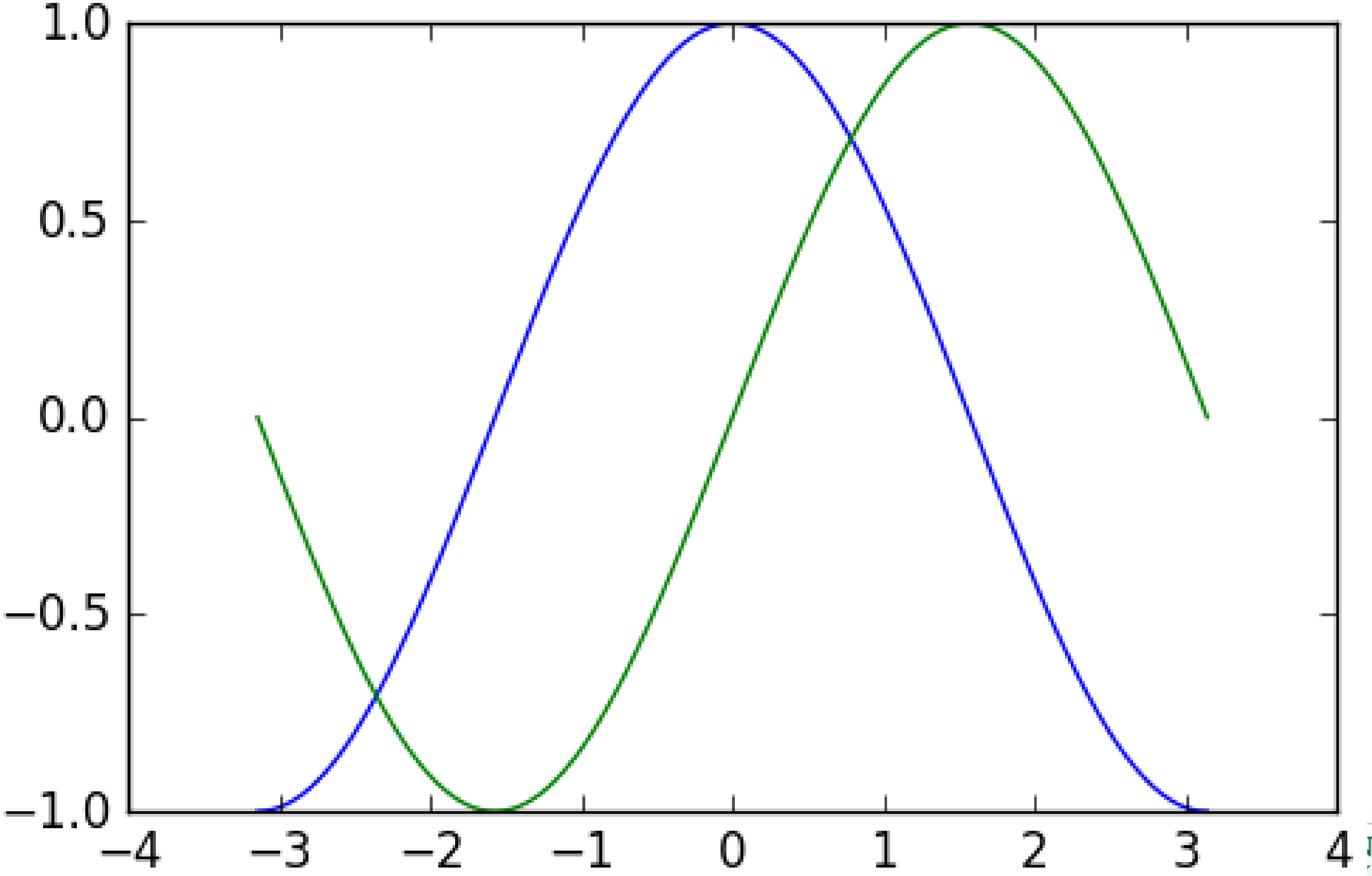


Plotting with Default Settings

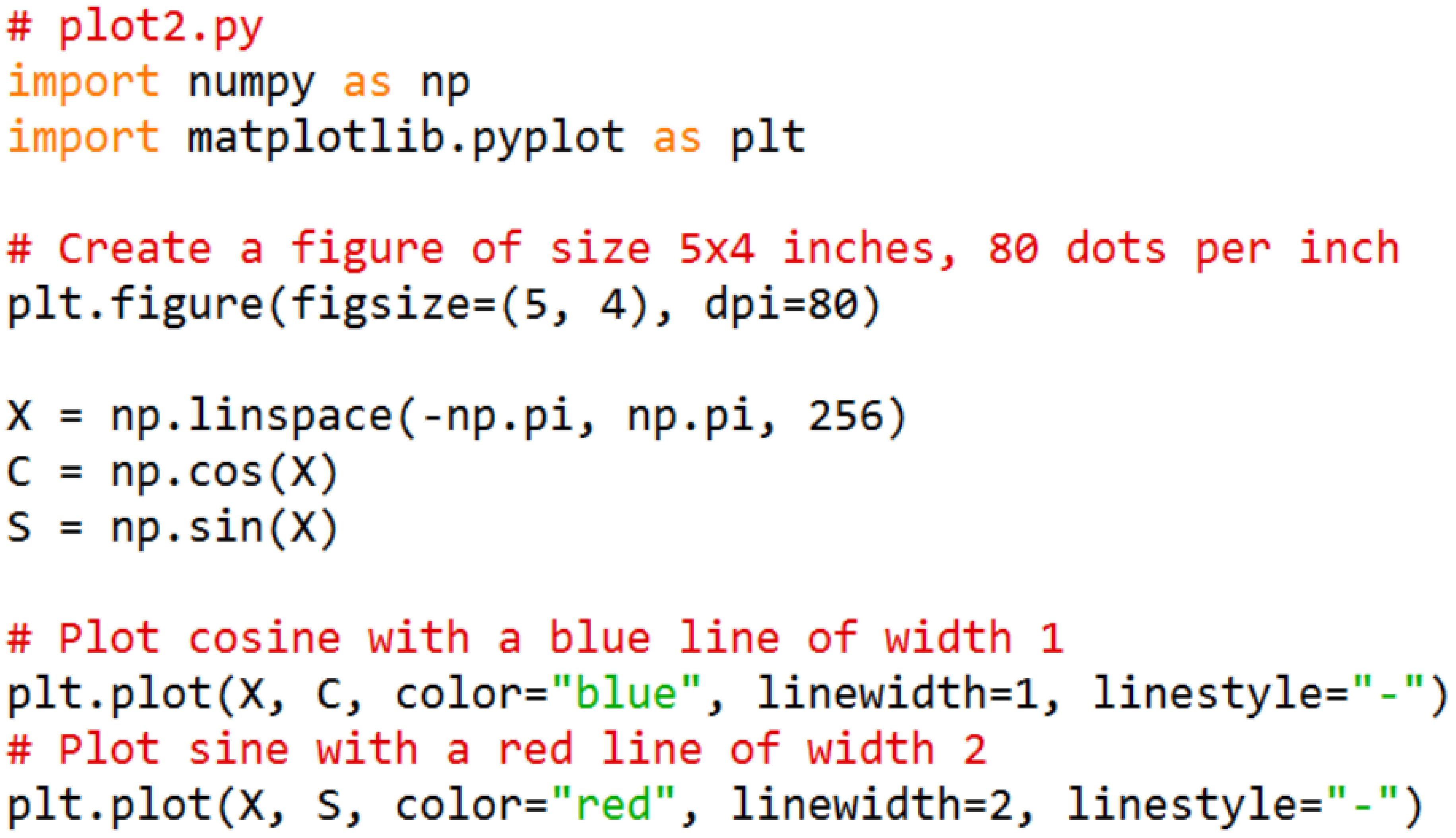


```
# plot1.py
import numpy as np
import matplotlib.pyplot as plt
X = np.linspace(-np.pi, np.pi, 256)
C = np.cos(X)
S = np.sin(X)
plt.plot(X, C)
plt.plot(X, S)
plt.show()
```

Plotting with Default Settings

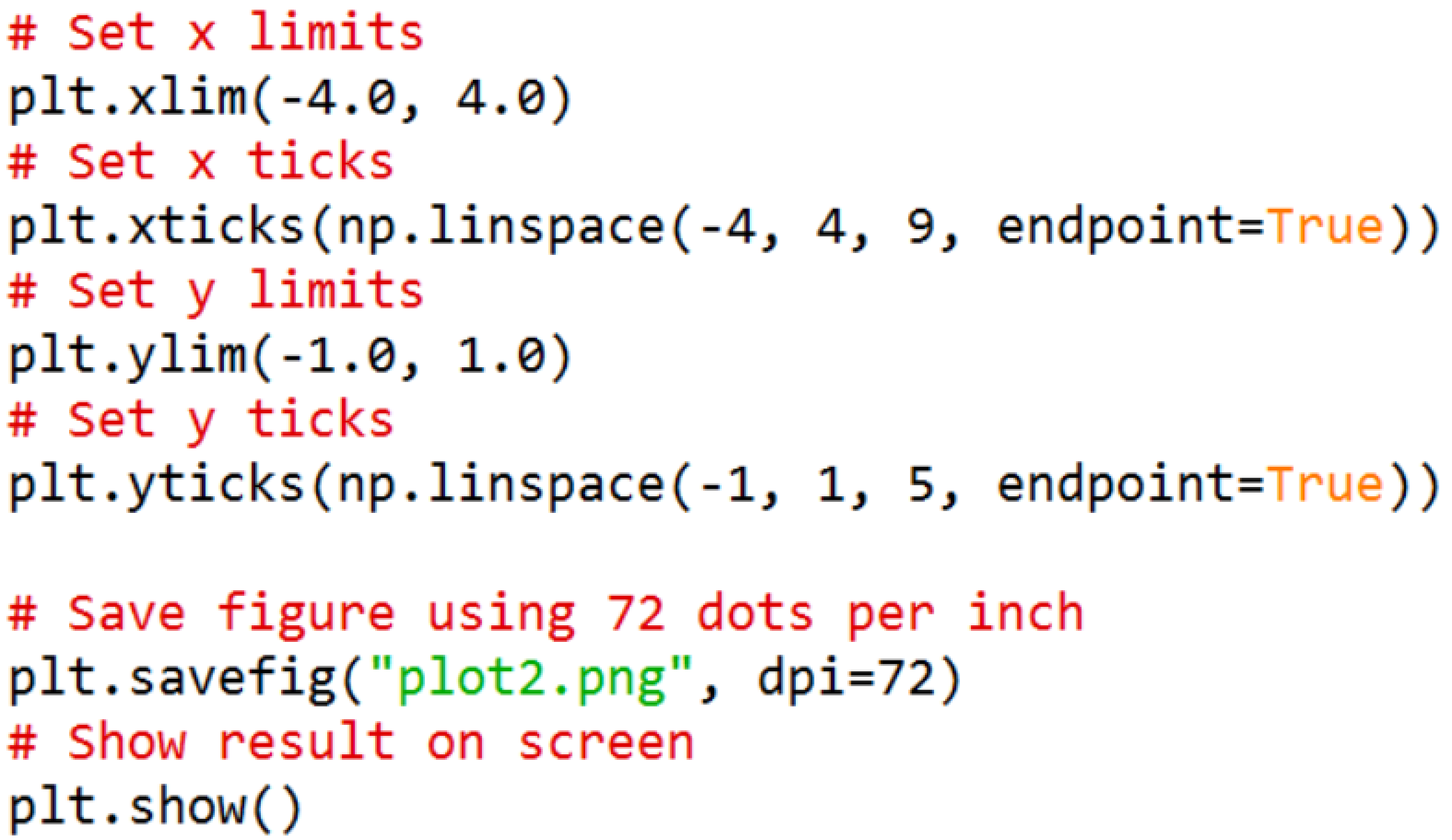


Setting Line Color and Width



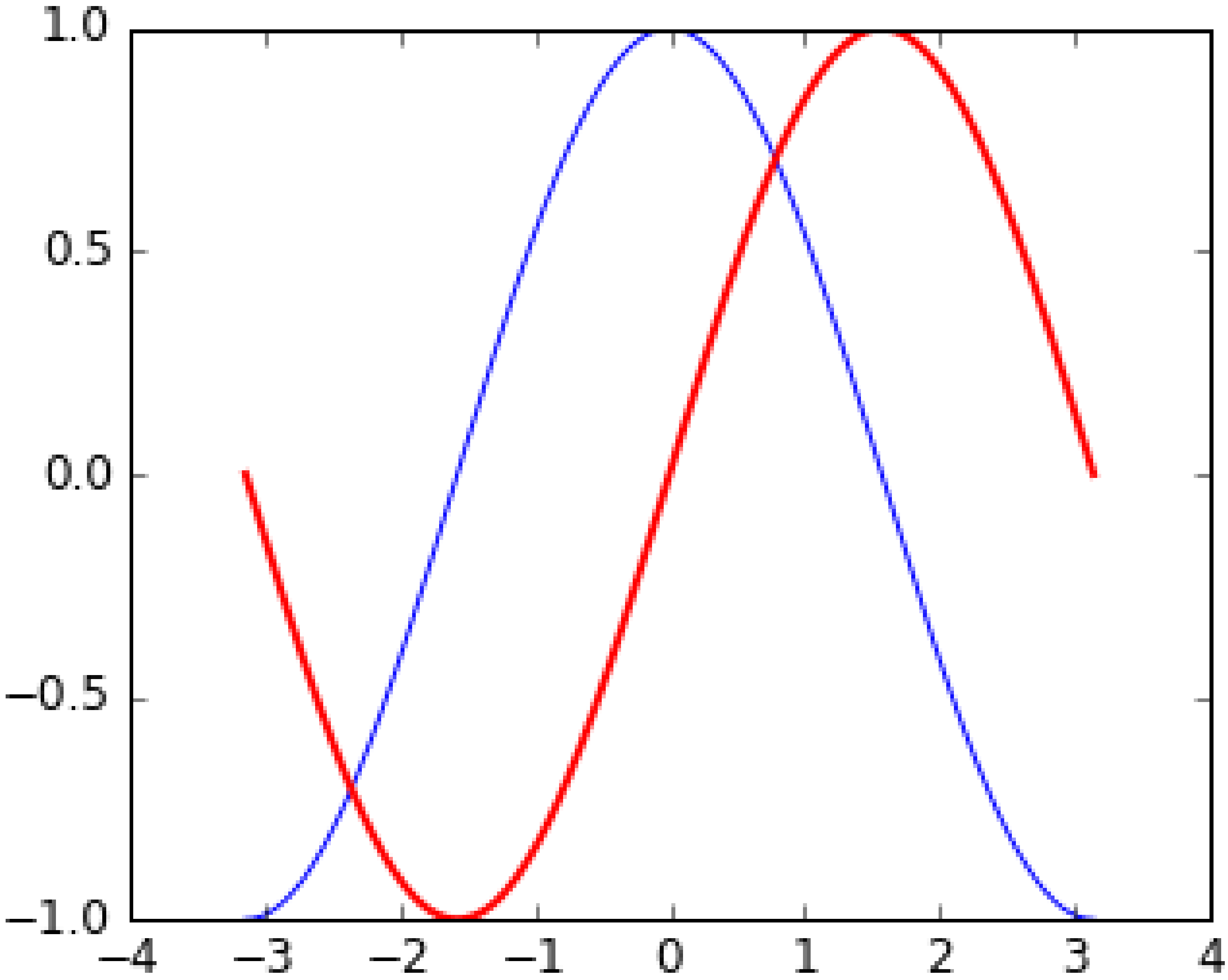
```
# plot2.py
import numpy as np
import matplotlib.pyplot as plt

# Create a figure of size 5x4 inches, 80 dots per inch
plt.figure(figsize=(5, 4), dpi=80)

X = np.linspace(-np.pi, np.pi, 256)
C = np.cos(X)
S = np.sin(X)

# Plot cosine with a blue line of width 1
plt.plot(X, C, color="blue", linewidth=1, linestyle="-")
# Plot sine with a red line of width 2
plt.plot(X, S, color="red", linewidth=2, linestyle="-")
```

Setting Limits and Save the Figure



```
# Set x limits
plt.xlim(-4.0, 4.0)
# Set x ticks
plt.xticks(np.linspace(-4, 4, 9, endpoint=True))
# Set y limits
plt.ylim(-1.0, 1.0)
# Set y ticks
plt.yticks(np.linspace(-1, 1, 5, endpoint=True))

# Save figure using 72 dots per inch
plt.savefig("plot2.png", dpi=72)
# Show result on screen
plt.show()
```

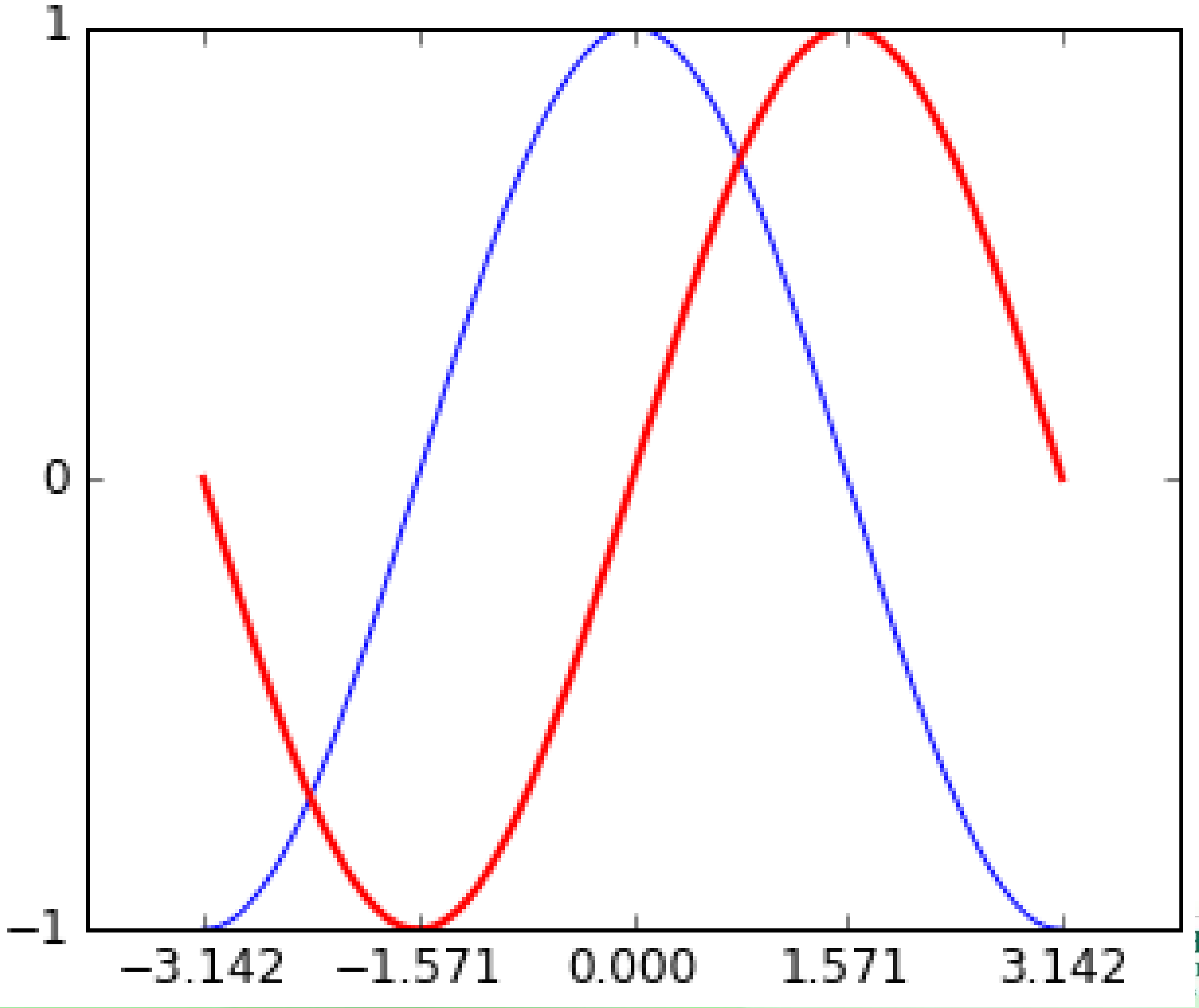
Setting Limits and Save the Figure



Setting Ticks

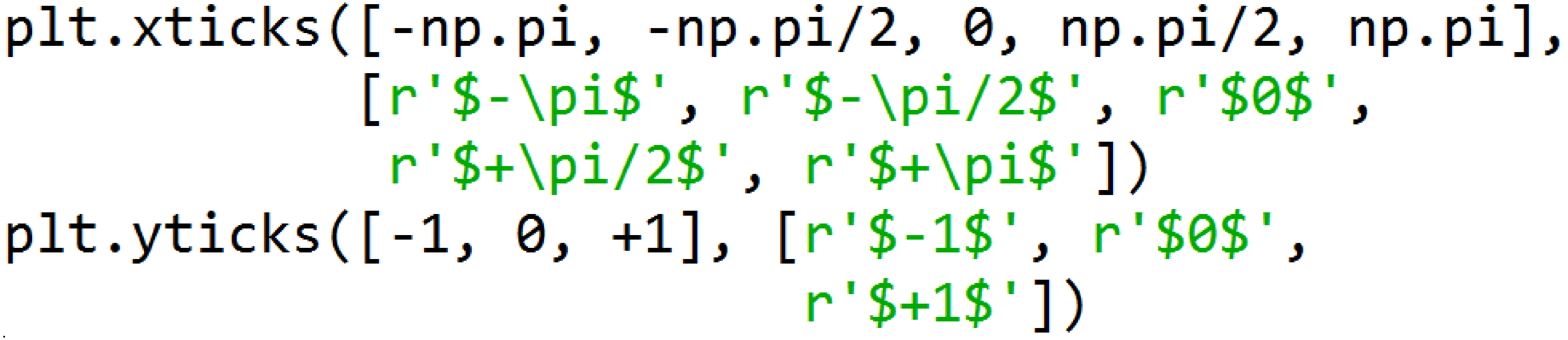
```
# plot3.py
...
# Set x limits
plt.xlim(-4.0, 4.0)
# Set x ticks
# Set x ticks to represent interesting values
# (+π, -π, +π/2, -π/2) for sine and cosine
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi])
# Set y limits
plt.ylim(-1.0, 1.0)
# Set y ticks
plt.yticks([-1, 0, +1])

...
```

Setting Ticks



Customize Tick Labels

```
# plot4.py
...
```

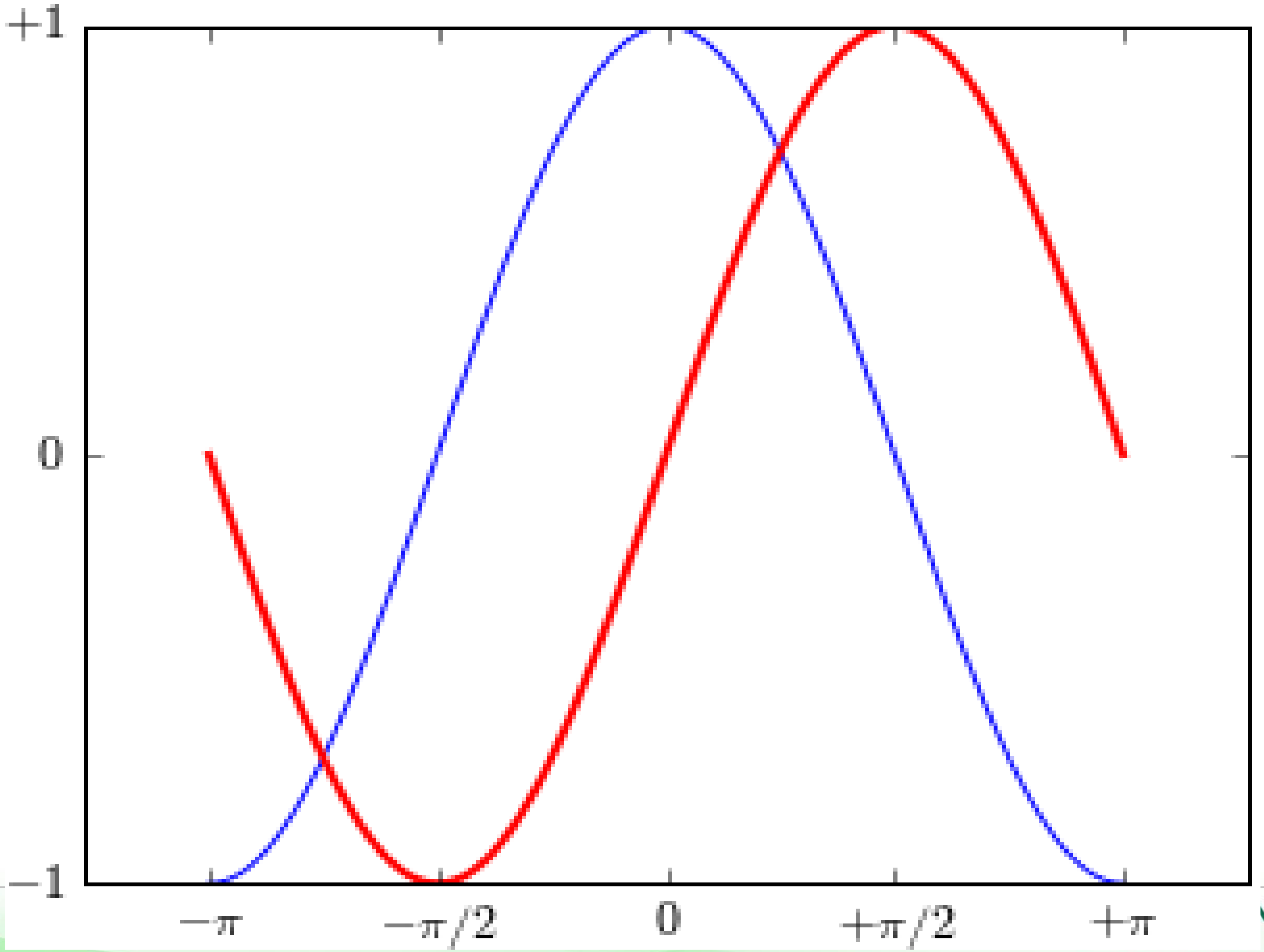


```
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi],
           [r'$-\pi$', r'$-\pi/2$', r'$0$',
            r'$+\pi/2$', r'$+\pi$'])
plt.yticks([-1, 0, +1], [r'$-1$', r'$0$',
                         r'$+1$'])
```

```
...
```

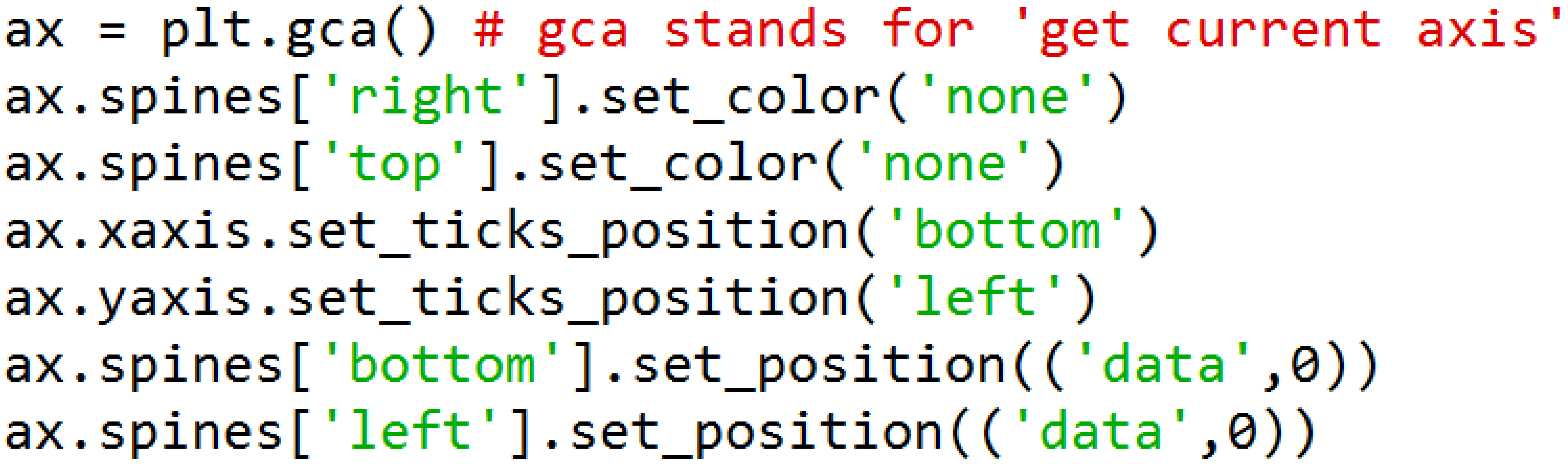
- Writing math expressions using LaTeX
  - Use raw strings (r)
  - Surround the math text with dollar signs ($)

Customize Tick Labels



Moving Spines

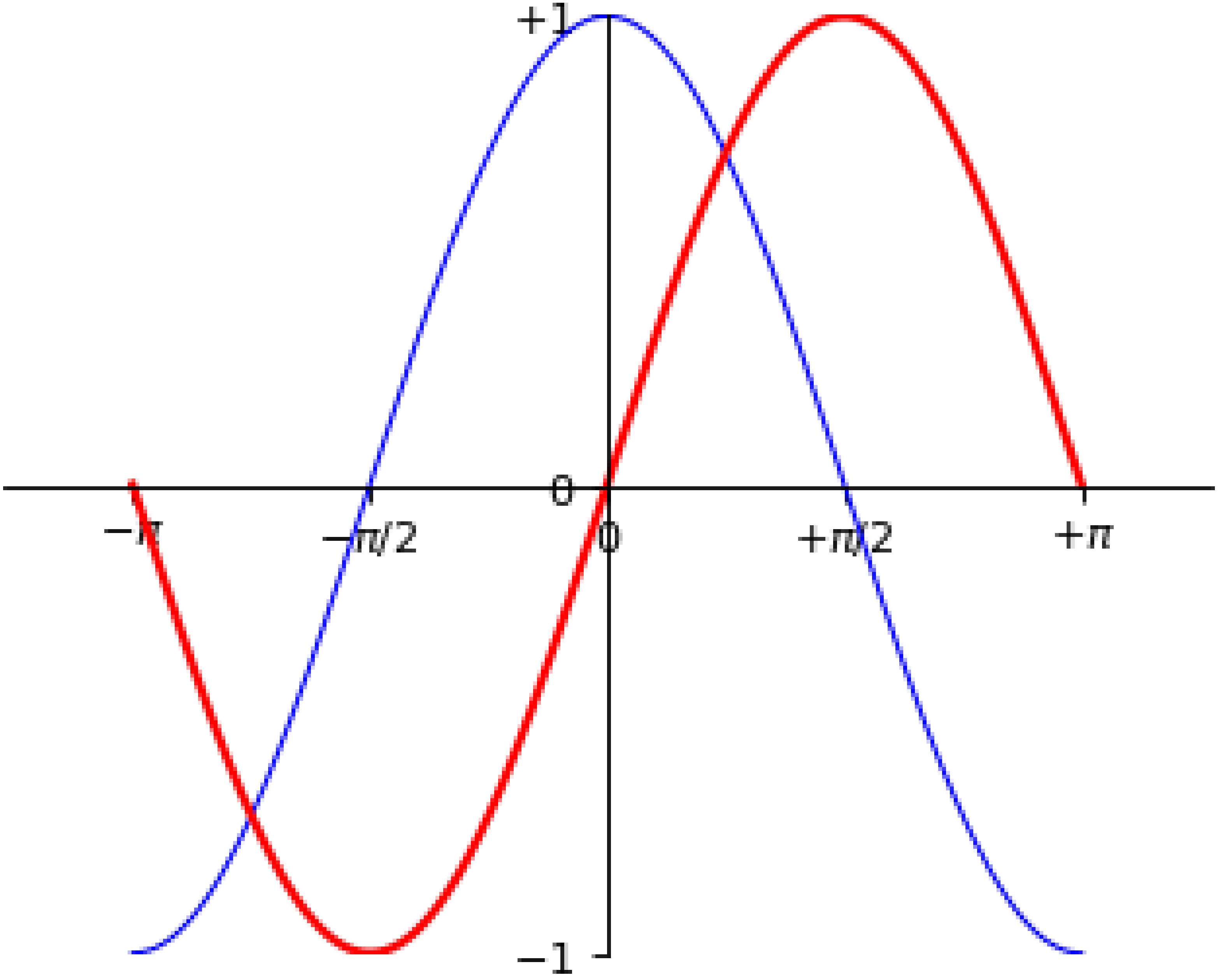
```
# plot5.py
...
```



```
ax = plt.gca() # gca stands for 'get current axis'
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')
ax.spines['bottom'].set_position(('data',0))
ax.spines['left'].set_position(('data',0))
```

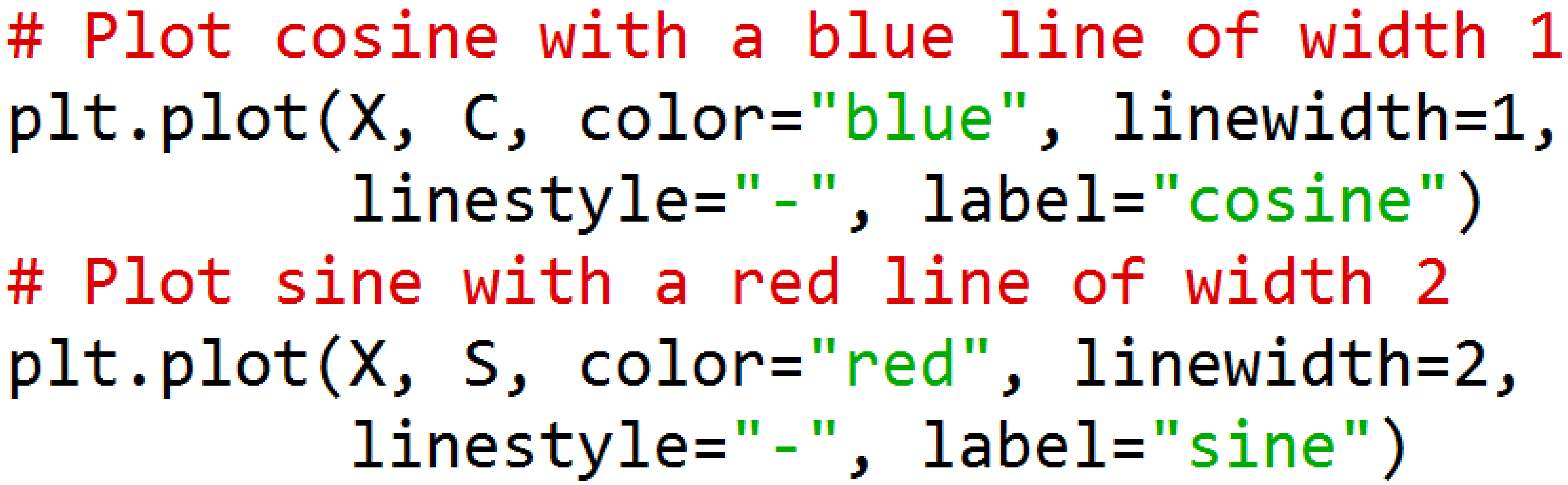
```
...
```

Moving Spines



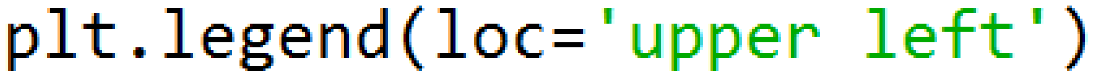
Adding A Legend

```
# plot6.py
...
```



```
# Plot cosine with a blue line of width 1
plt.plot(X, C, color="blue", linewidth=1,
         linestyle="-", label="cosine")
# Plot sine with a red line of width 2
plt.plot(X, S, color="red", linewidth=2,
         linestyle="-", label="sine")
```

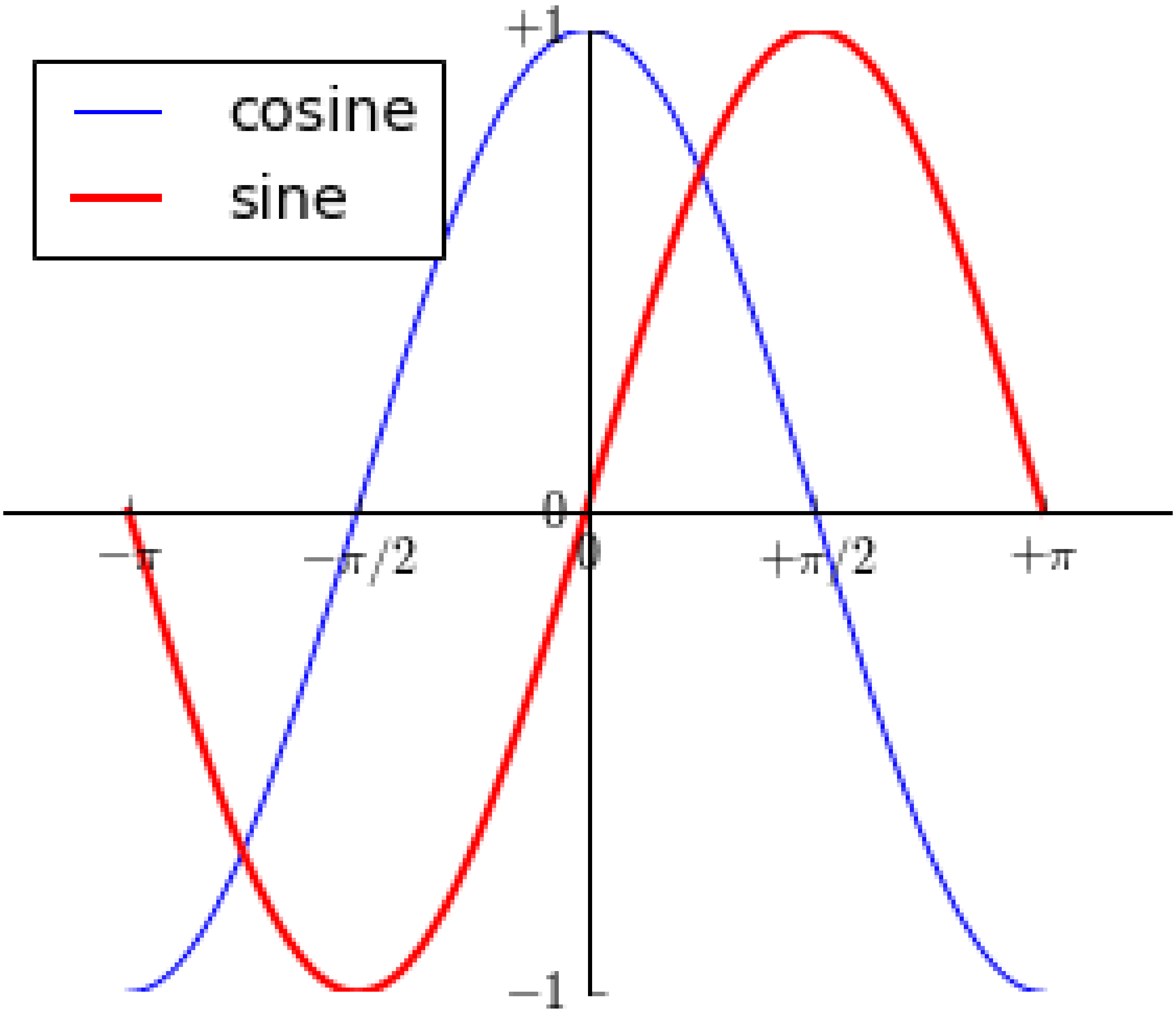
```
...
```



```
plt.legend(loc='upper left')
```

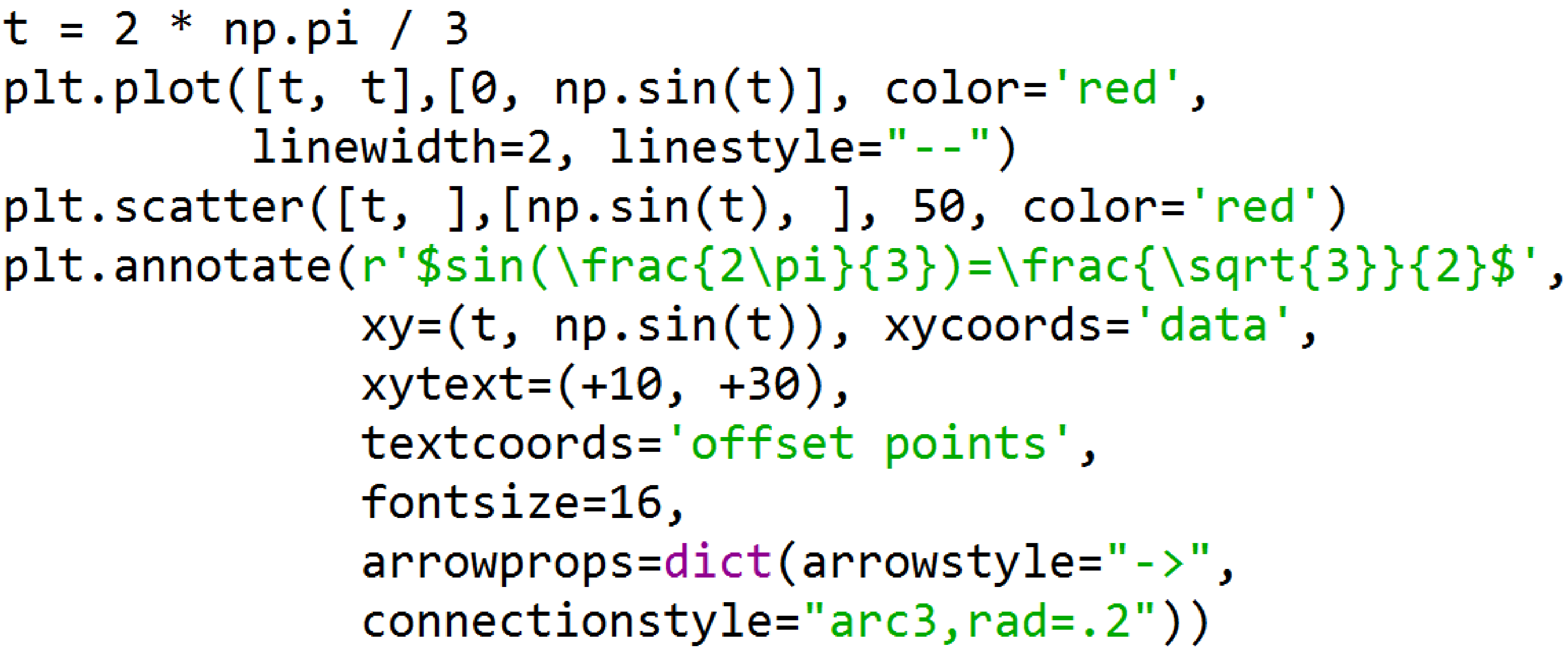
```
...
```

Adding A Legend



[Annotate](#) Some Points

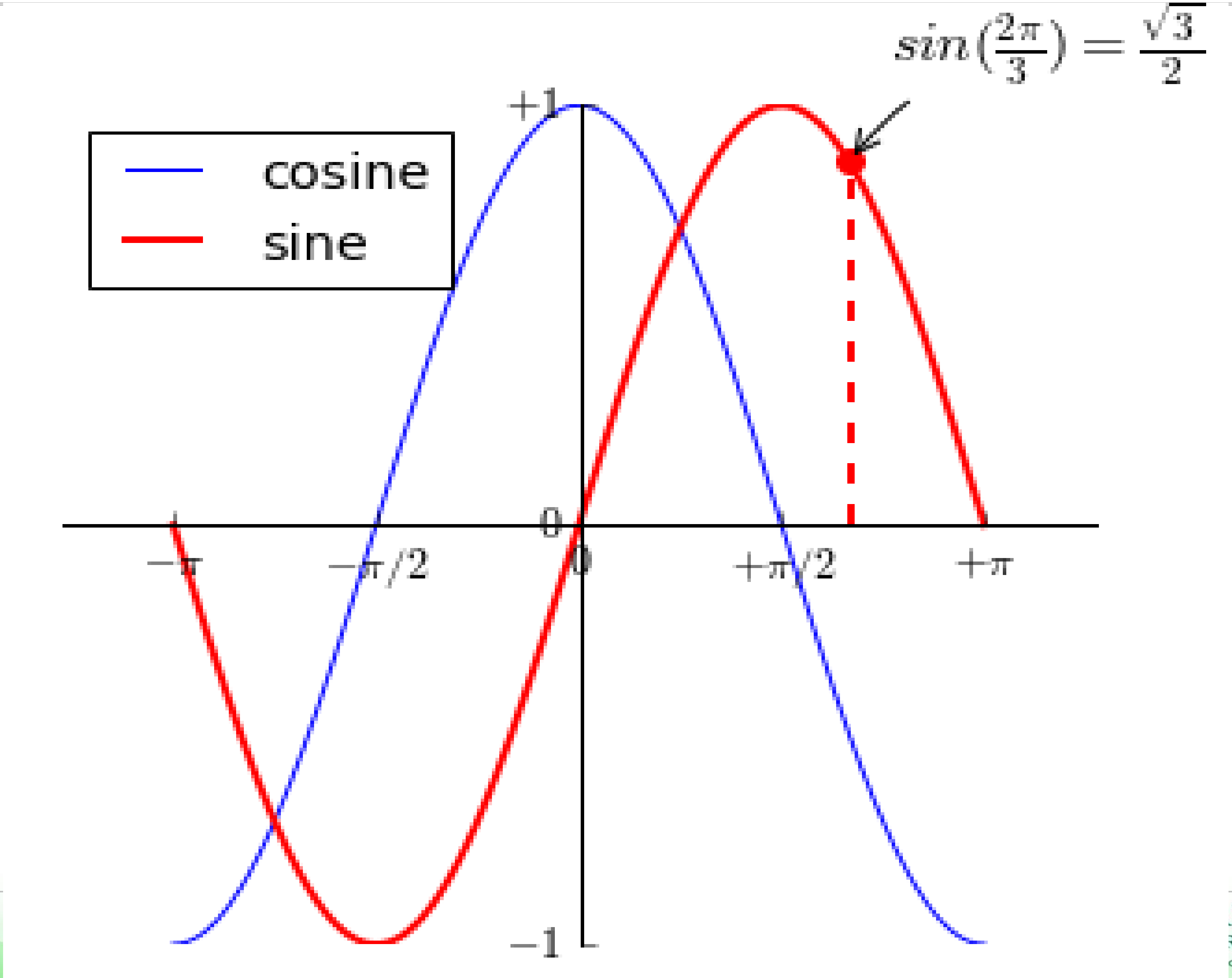
```
# plot7.py
```



```
t = 2 * np.pi / 3
plt.plot([t, t],[0, np.sin(t)], color='red',
         linewidth=2, linestyle="--")
plt.scatter([t, ],[np.sin(t), ], 50, color='red')
plt.annotate(r'$sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
             xy=(t, np.sin(t)), xycoords='data',
             xytext=(+10, +30),
             textcoords='offset points',
             fontsize=16,
             arrowprops=dict(arrowstyle="->",
             connectionstyle="arc3,rad=.2"))
```

```
...
```

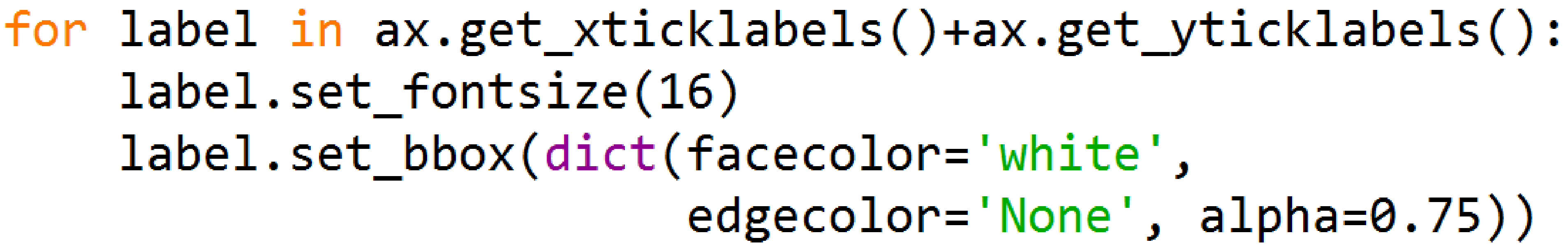
Annotate Some Points



Refine the Details

- Make the tick labels bigger
- Make them semi-transparent with white background

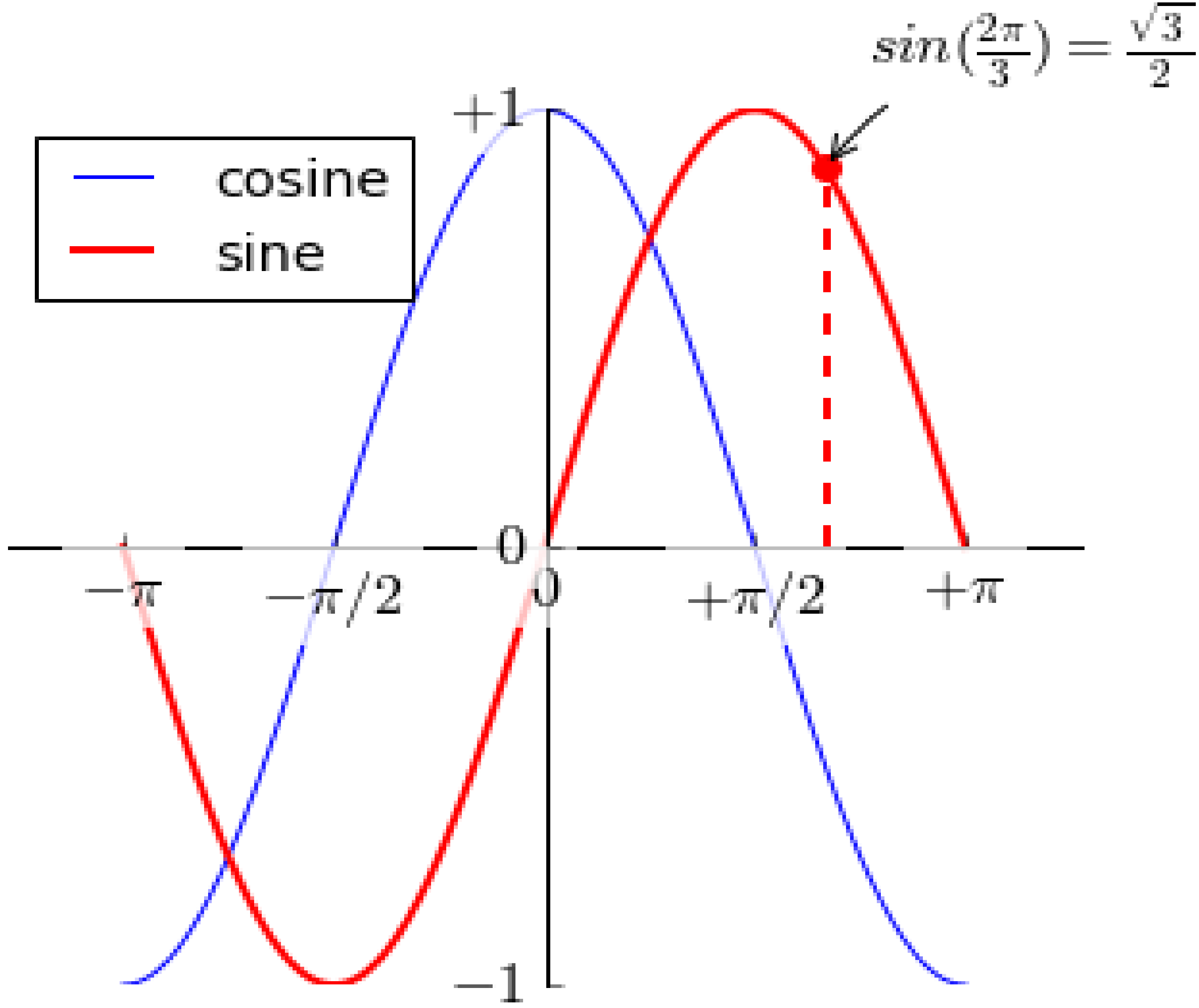
```
# plot8.py
...
```



```
for label in ax.get_xticklabels()+ax.get_yticklabels():
    label.set_fontsize(16)
    label.set_bbox(dict(facecolor='white',
                        edgecolor='None', alpha=0.75))
```

```
...
```

Refine the Details


$sin(\frac{2\pi}{3}) = \frac{\sqrt{3}}{2}$
cosine
sine
$-\pi$
$-\pi/2$
$+\pi/2$
$+\pi$