

**" USE THE MATPLOTLIB, LUKE "**

WEN-WEI LIAO  
gattacaliao@gmail.com

A long time ago, in a galaxy far, far away ...



**John Hunter**

“

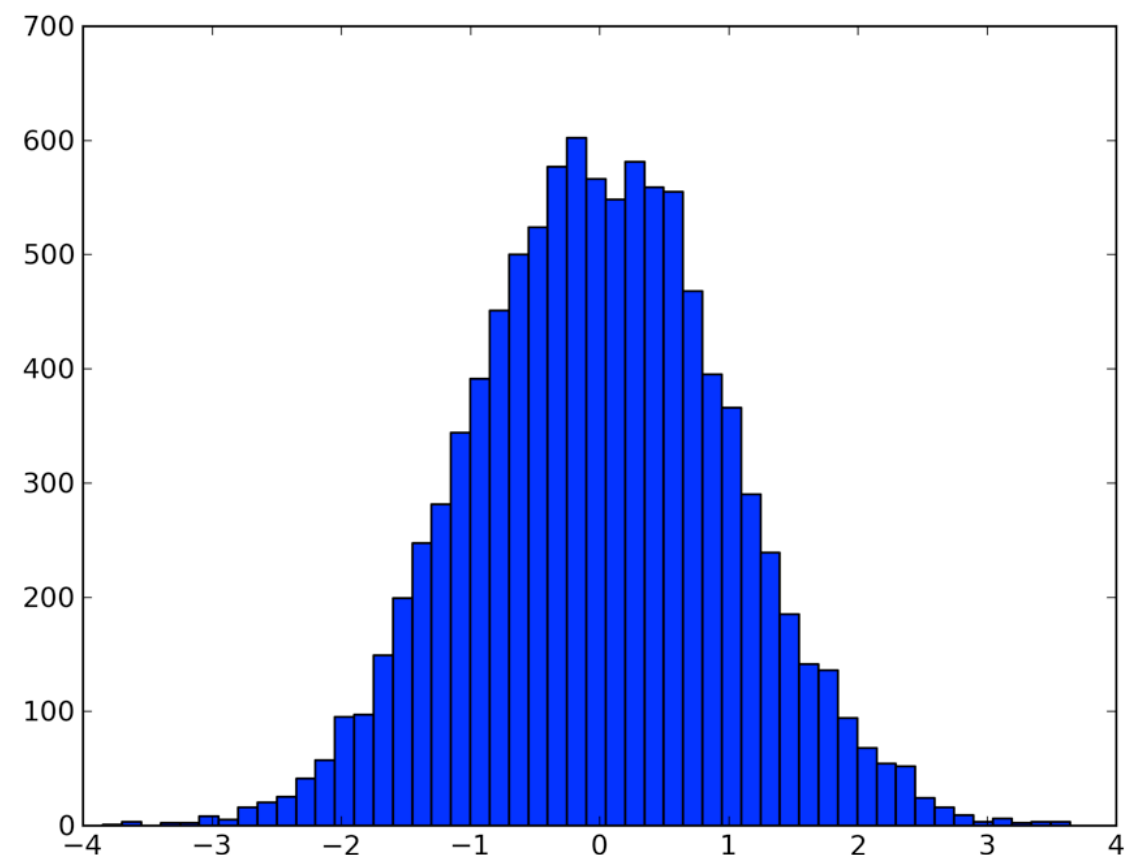
**Matplotlib** is a Python 2D plotting package which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

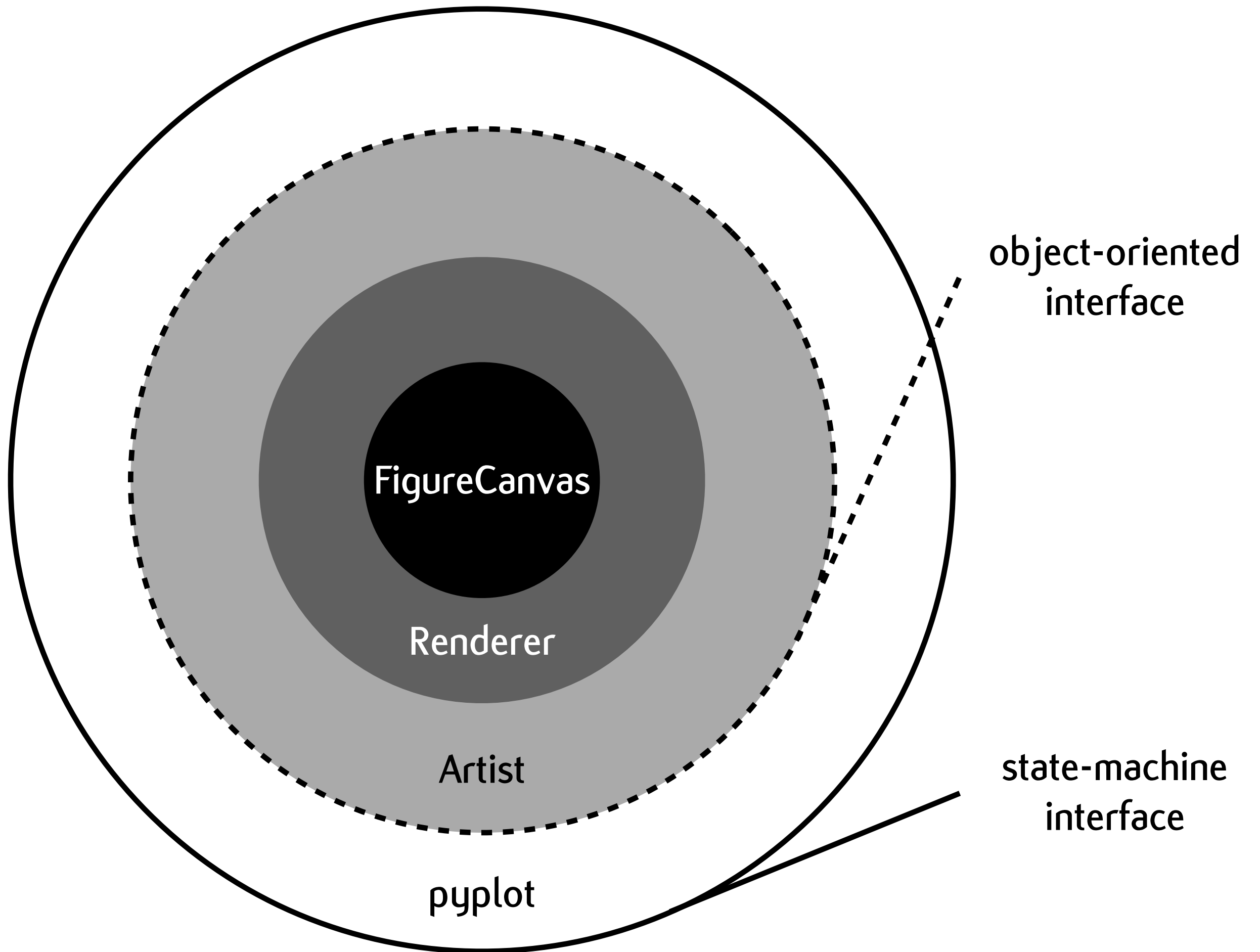
# Philosophy

create simple plots with just a few commands, or just one!

```
import numpy as np
import matplotlib.pyplot as plt

x = np.random.randn(10000)
plt.hist(x, bins=50)
plt.show()
```





**pyplot** provides a MATLAB-style state-machine interface to the underlying object-oriented interface in matplotlib

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 10, 0.1)
y = np.sin(x)
plt.plot(x, y)
plt.show()
```

**pylab** lumps pyplot together with numpy in a single namespace, making that environment even more MATLAB-like

```
from pylab import *

x = arange(0, 10, 0.1)
y = sin(x)
plot(x, y)
show()
```

Explicit is better than implicit.



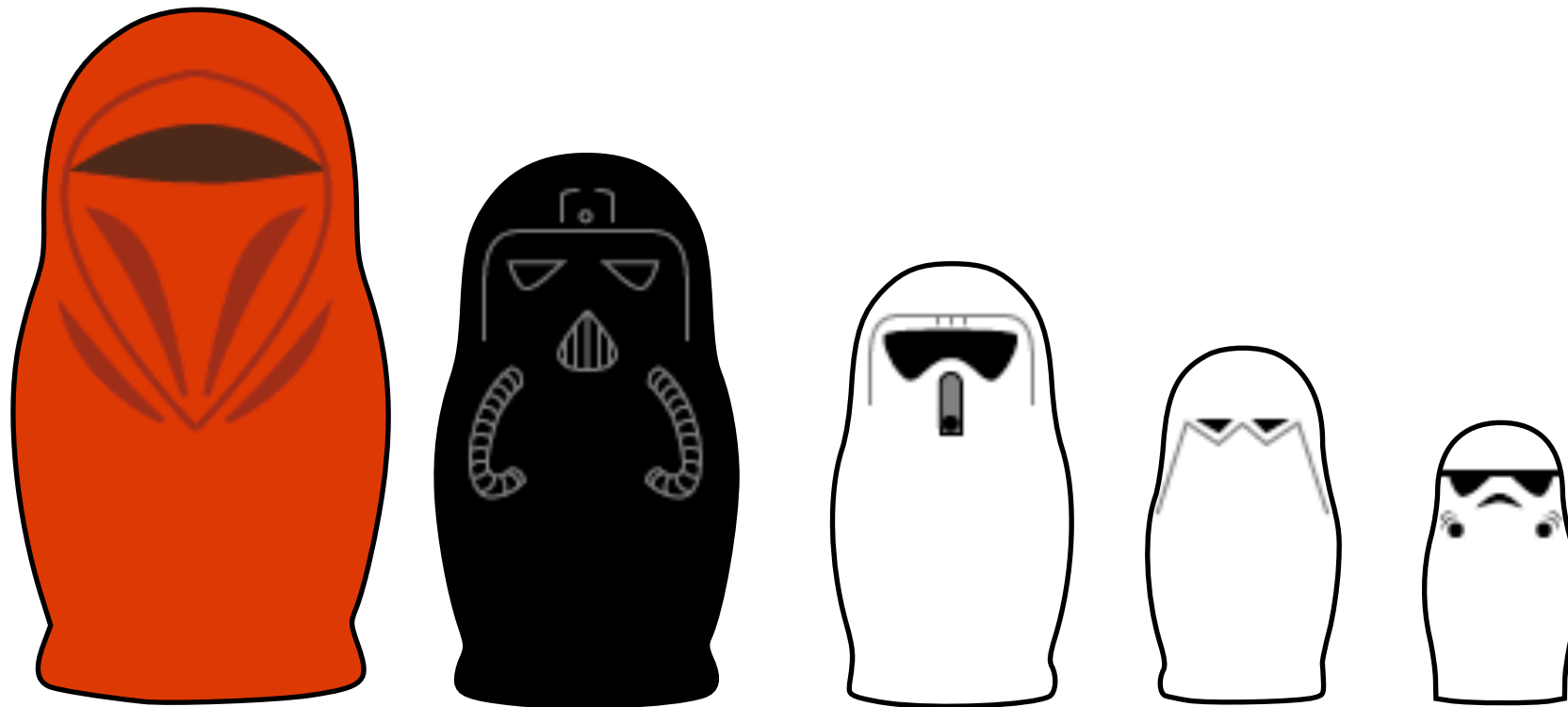
**preferred style** using pyplot convenience functions,  
but object-orientation for the rest

```
import numpy as np
import matplotlib.pyplot as plt

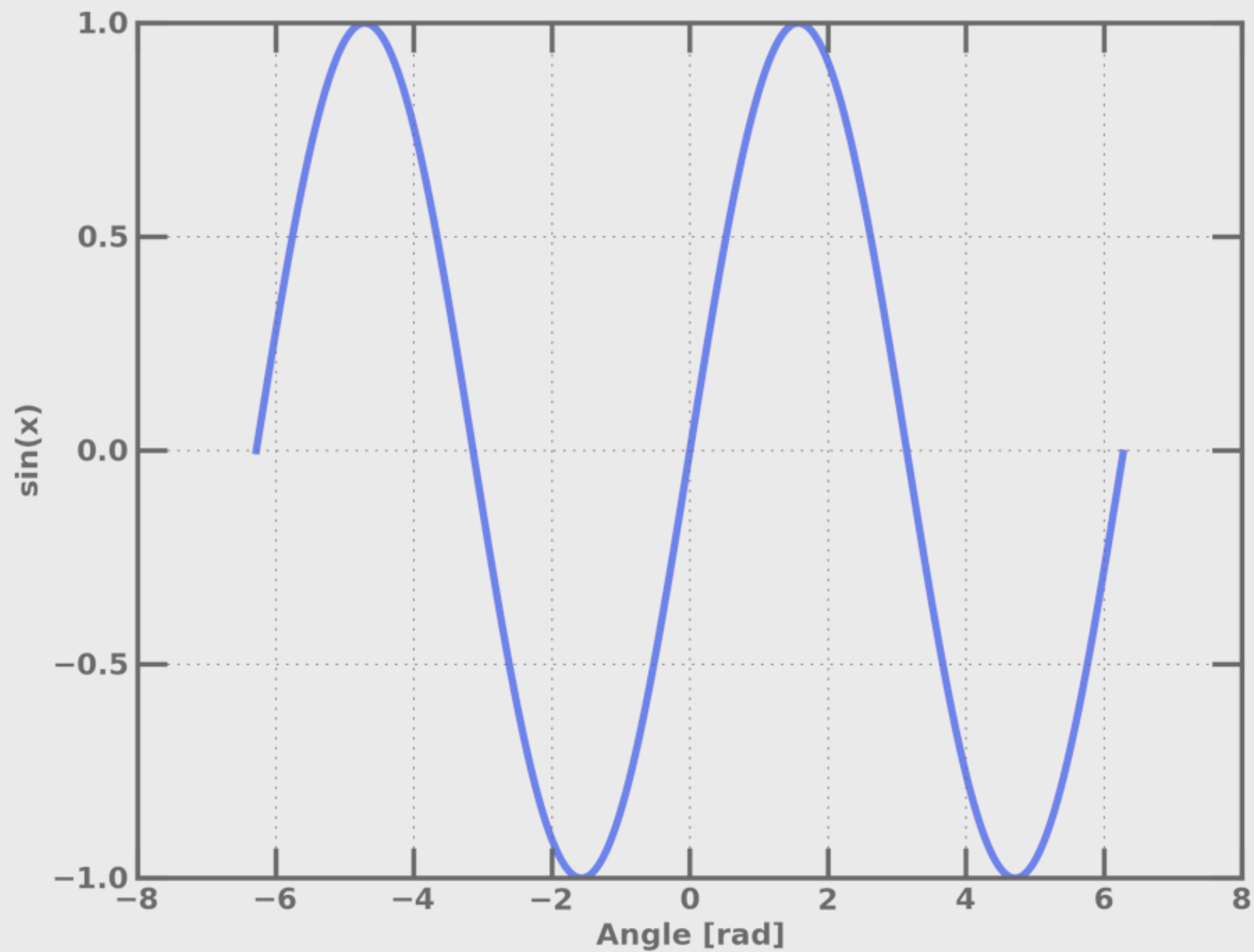
x = np.arange(0, 10, 0.1)
y = np.sin(x)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(x, y)
plt.show()
```

## 2 types of Artists

- Primitives: Line2D, Rectangle, Text, etc.
- Containers: Figure, Axes, Axis, Tick

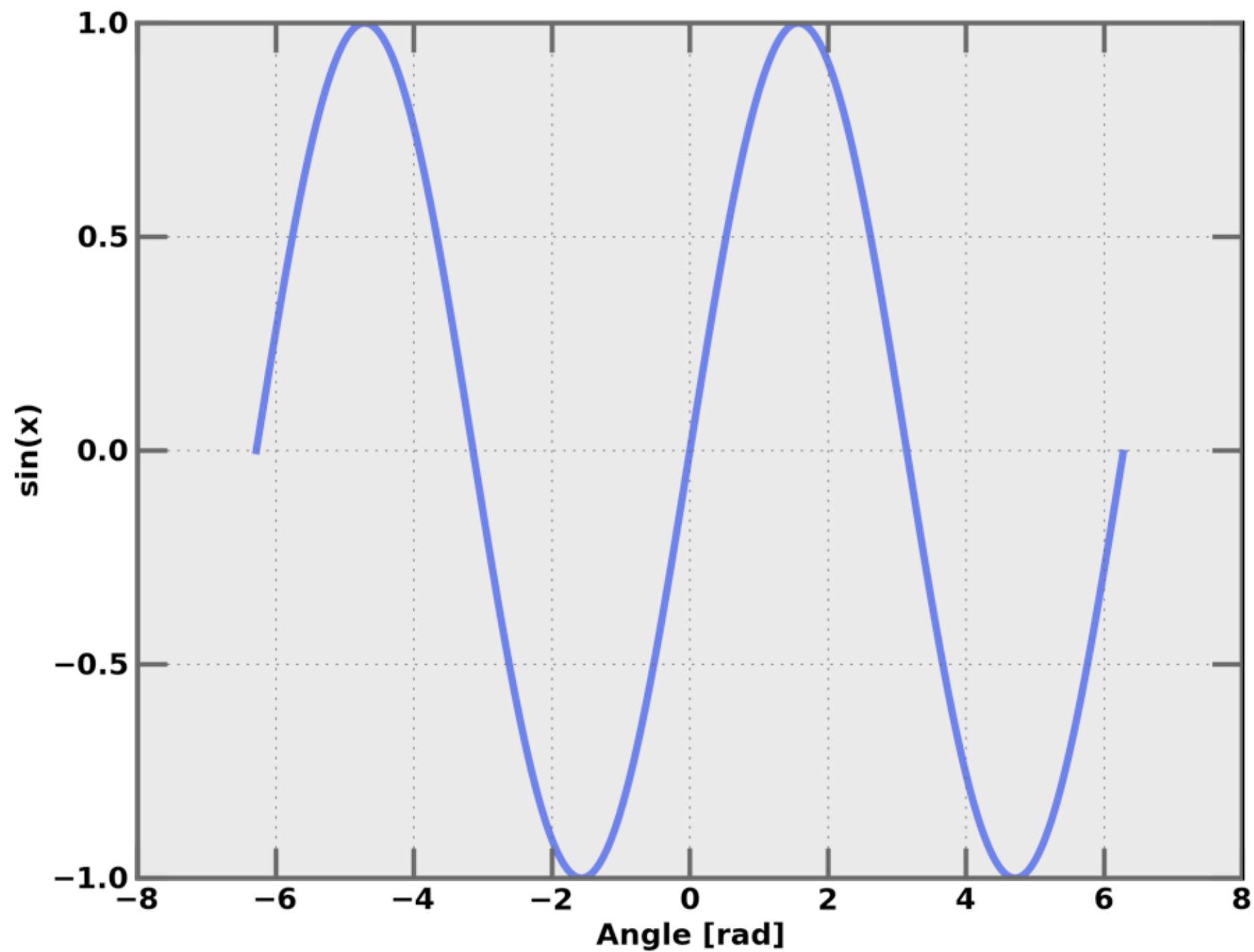


**Figure Container**  
(matplotlib.figure.Figure)



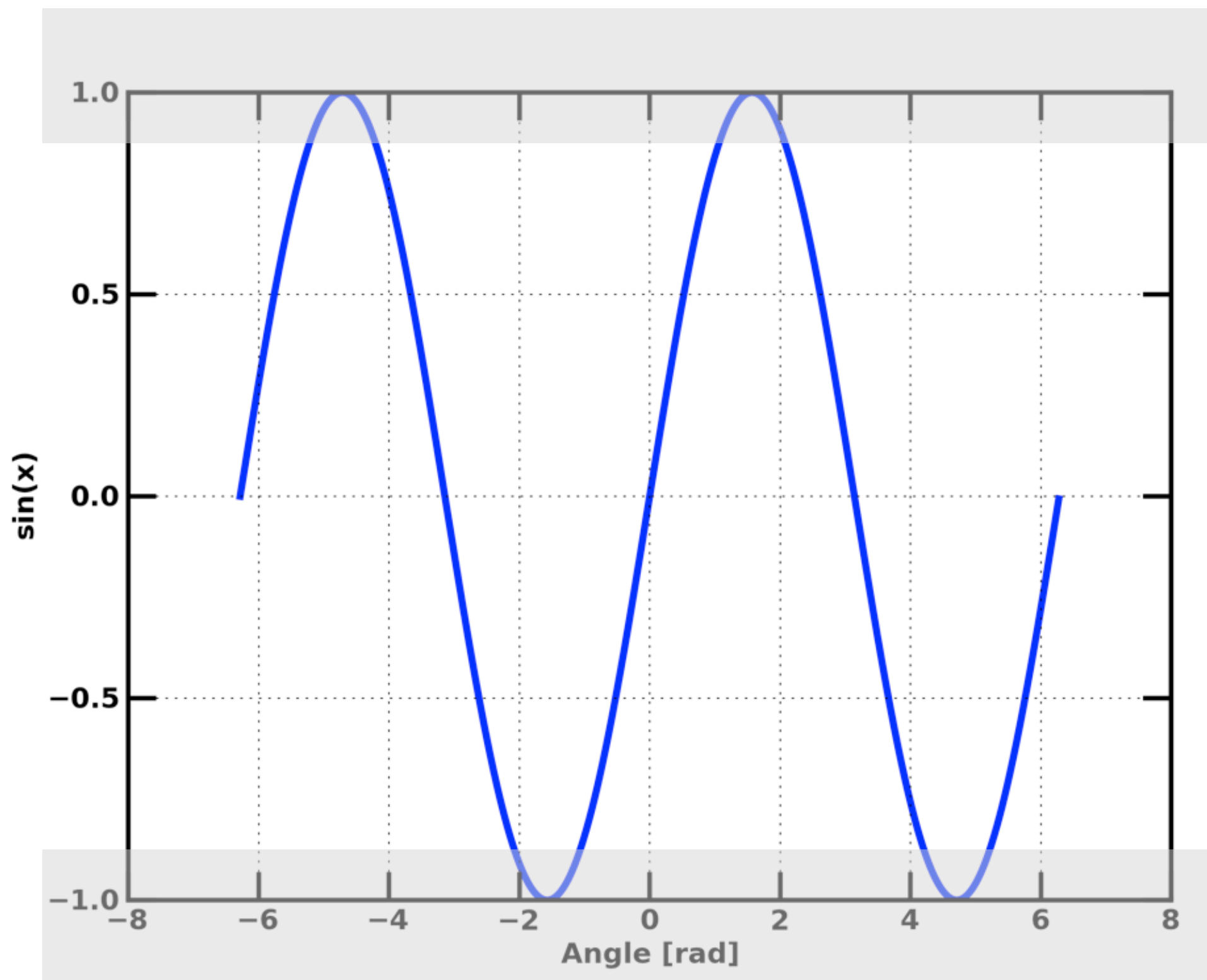
# Axes Container

(matplotlib.axes.Axes)



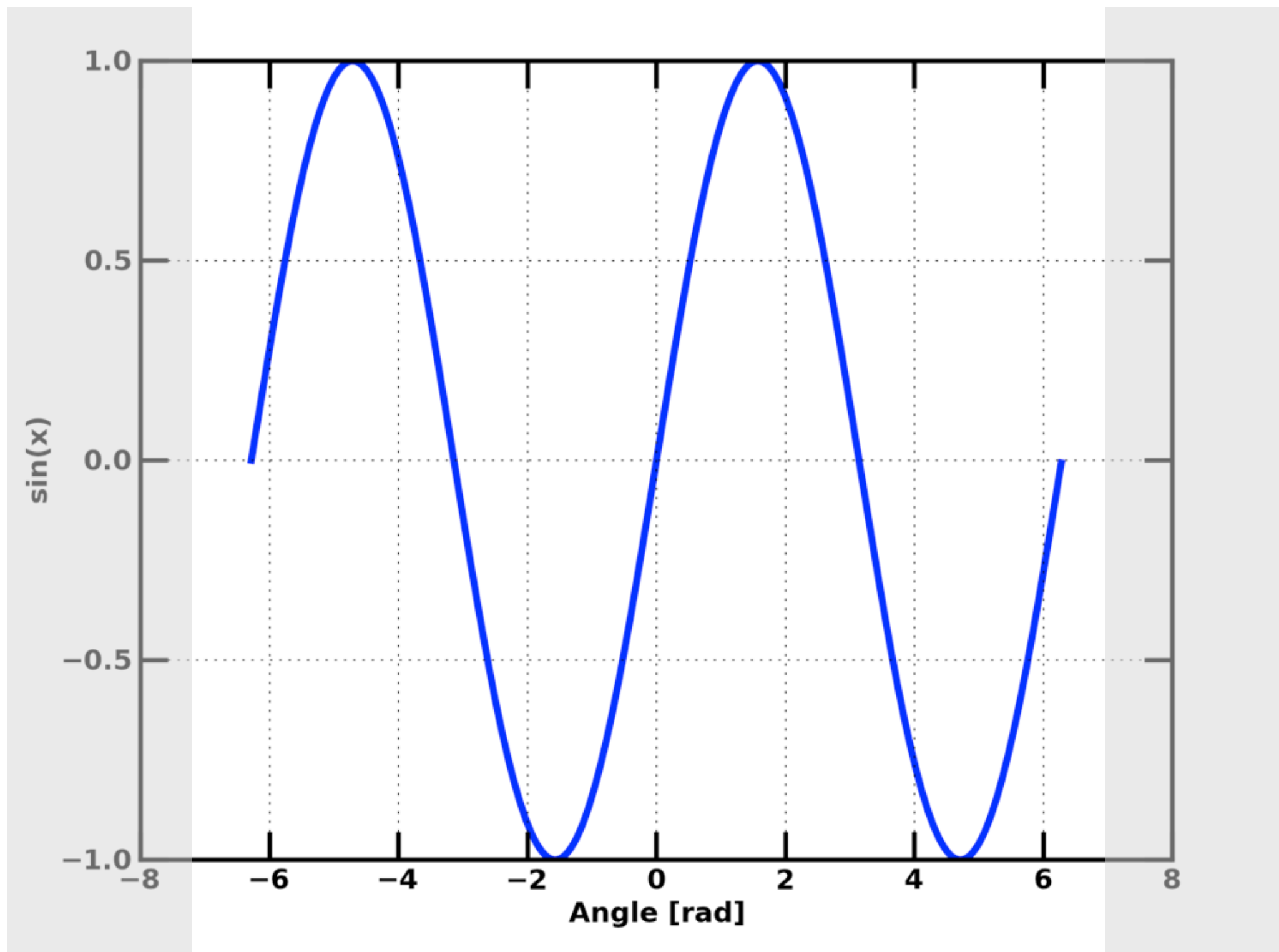
# XAxis Container

(matplotlib.axis.Axis)



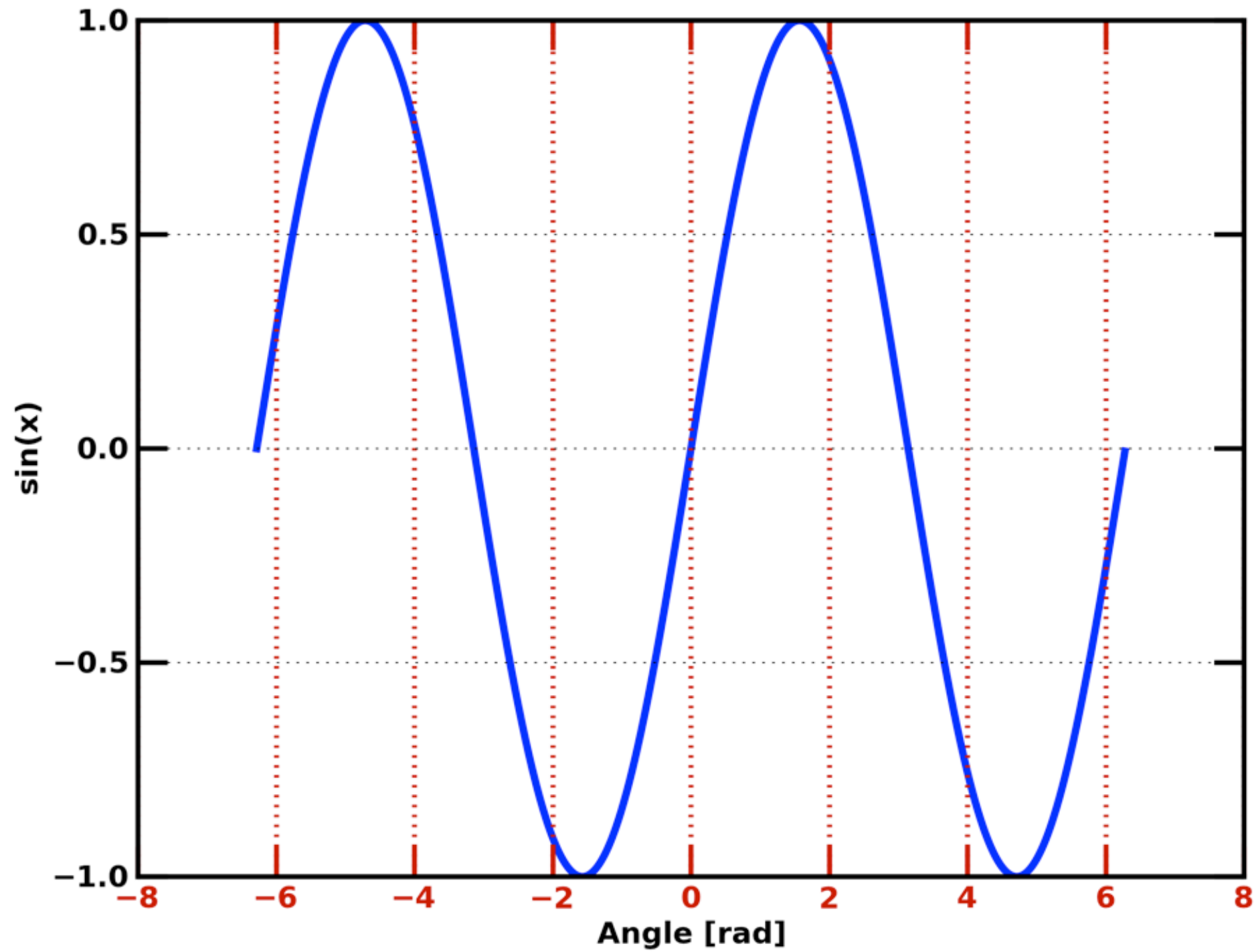
# YAxis Container

(matplotlib.axis.Axis)



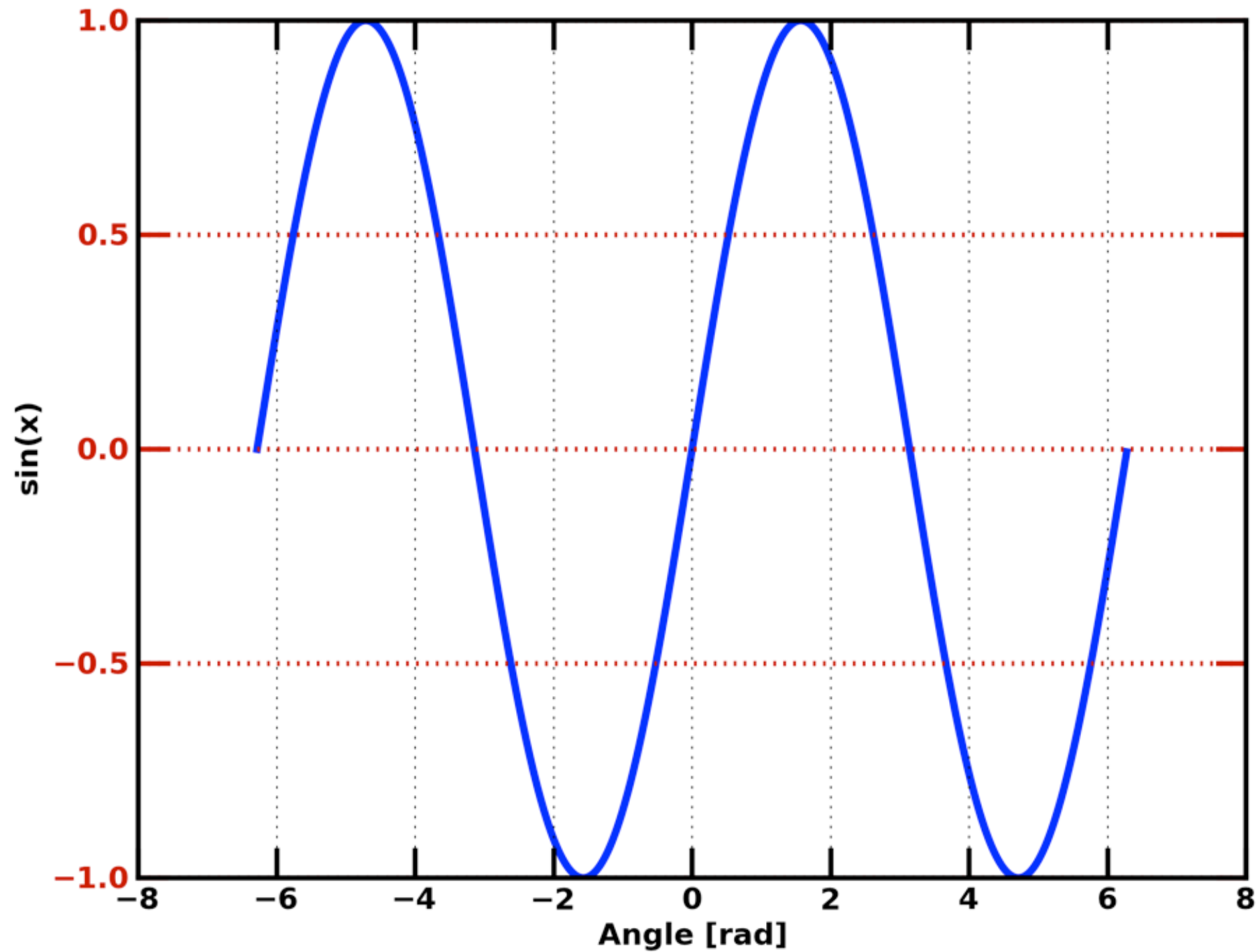
# XTick Container

(matplotlib.axis.Tick)



# YTick Container

(matplotlib.axis.Tick)





# Customizing your objects

each of the properties is accessed with an old-fashioned setter or getter

```
a = o.get_alpha()  
o.set_alpha(0.5*a)
```

set a number of properties at once

```
o.set(alpha=0.5, zorder=2)
```

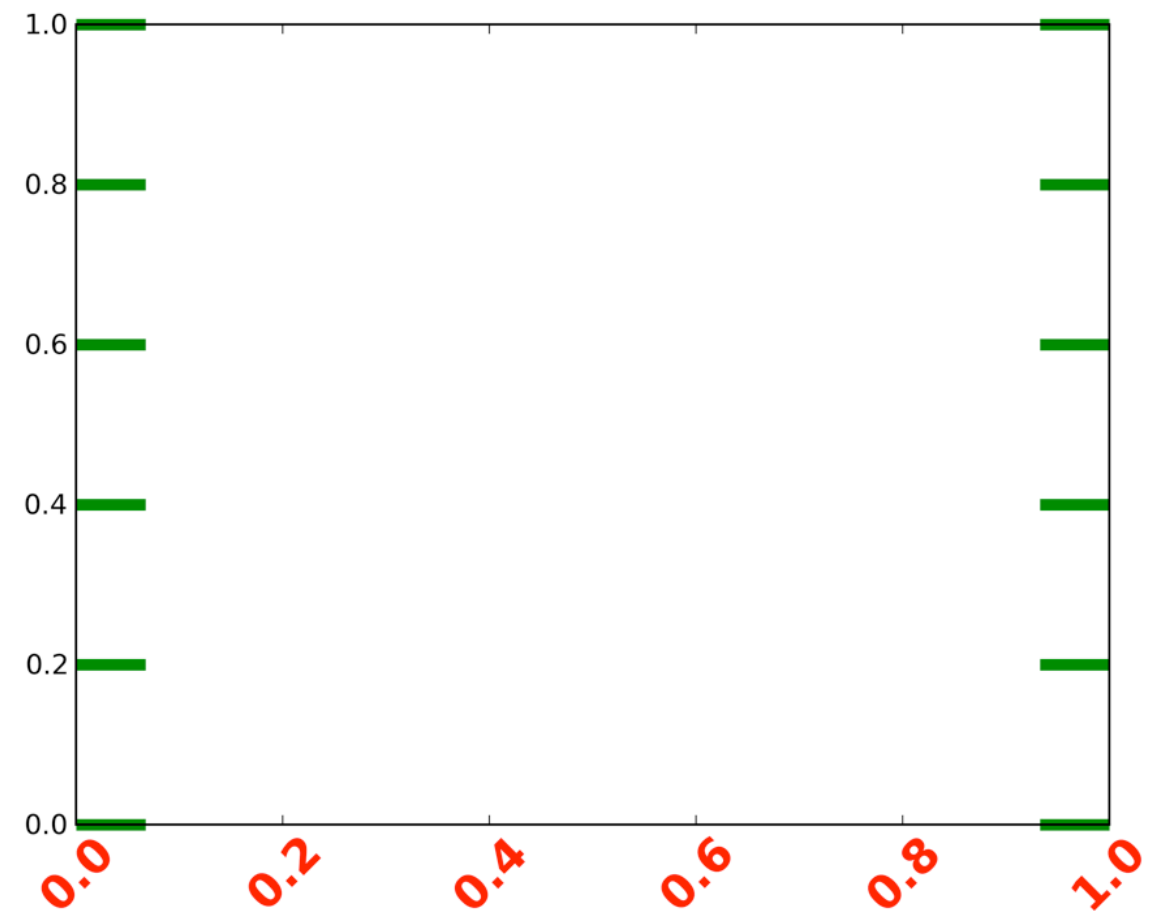
```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111)

for label in ax.xaxis.get_ticklabels():
    # label is a Text instance
    label.set_color('red')
    label.set_rotation(45)
    label.set_fontsize(20)
    label.set_fontweight('bold')

for line in ax.yaxis.get_ticklines():
    # line is a Line2D instance
    line.set_color('green')
    line.set_markersize(30)
    line.set_markeredgewidth(5)

plt.show()
```



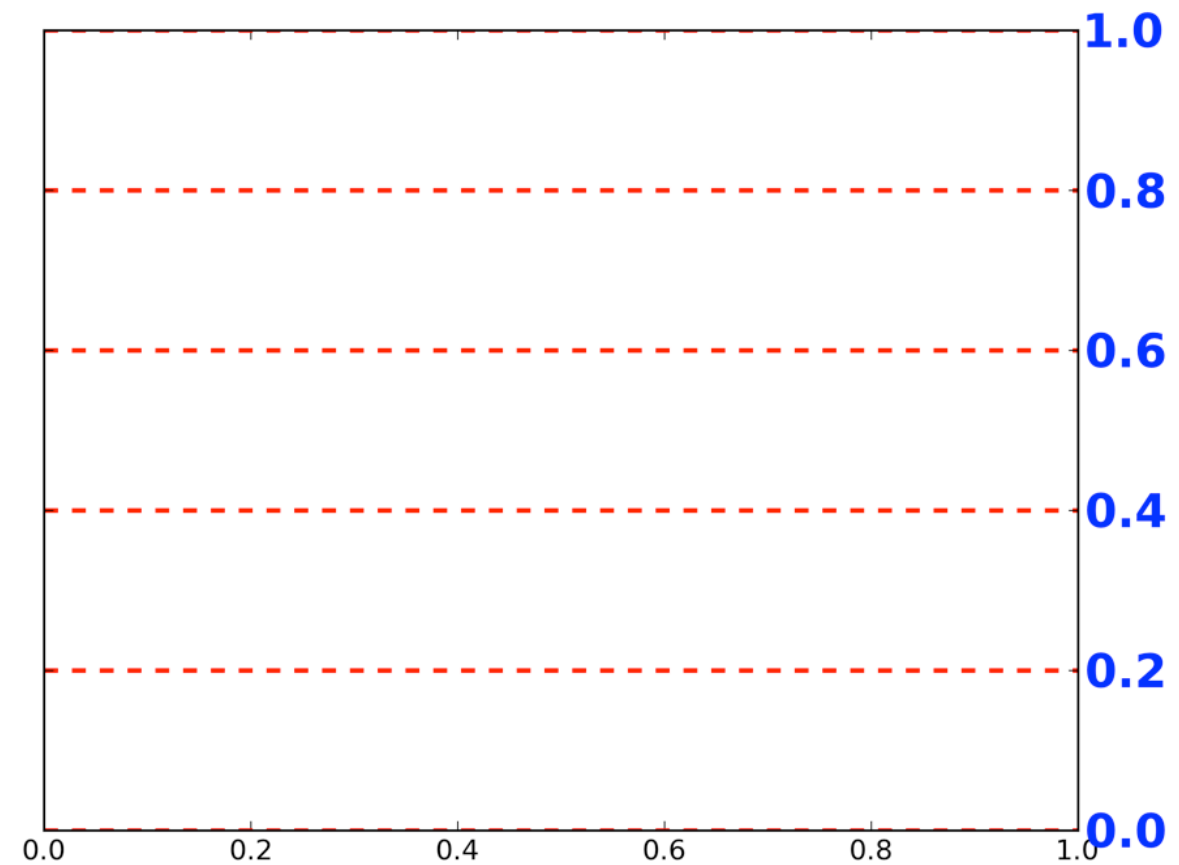
```
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111)

for tick in ax.yaxis.get_major_ticks():
    tick.label10n = False
    tick.label20n = True
    tick.label2.set_color('blue')
    tick.label2.set_fontsize(20)
    tick.label2.set_fontweight('bold')

    tick.grid0n = True
    tick.gridline.set_color('red')
    tick.gridline.set_linewidth(2)
    tick.gridline.set_linestyle('--')

plt.show()
```



```
import matplotlib.pyplot as plt

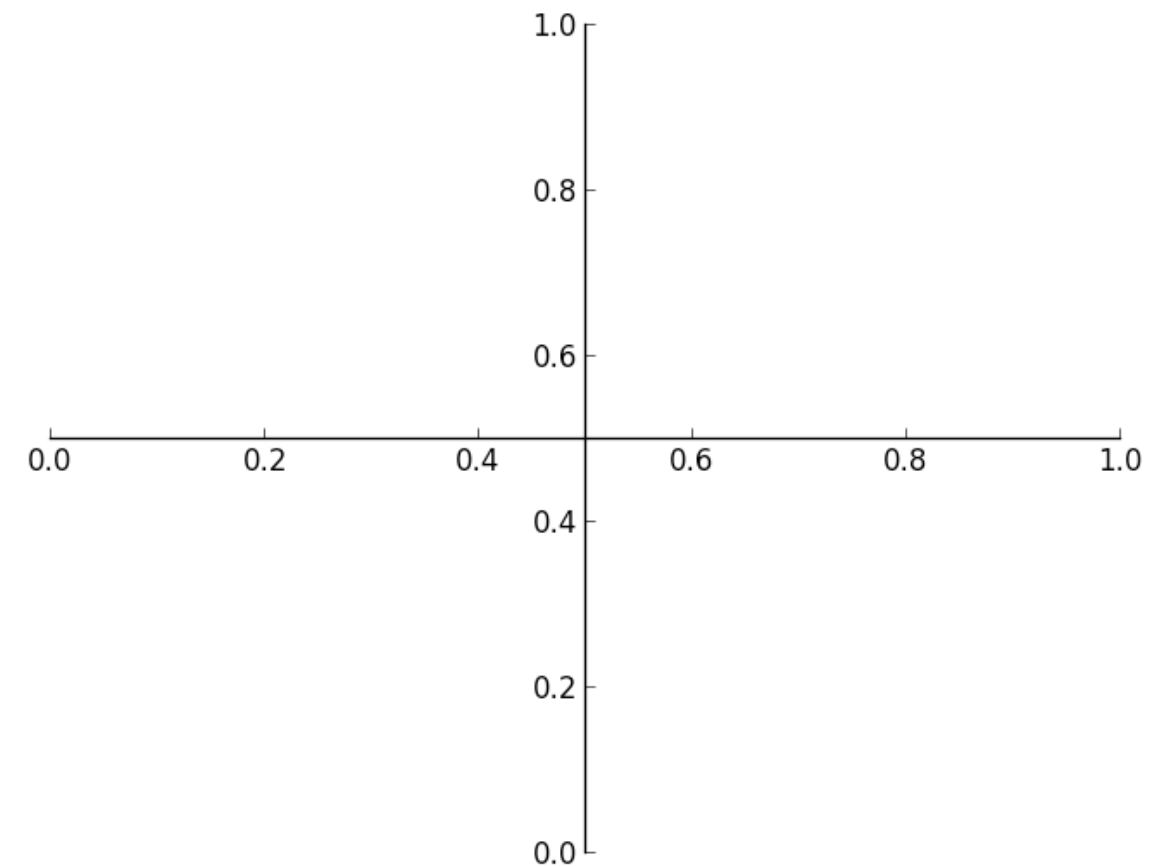
fig = plt.figure()
ax = fig.add_subplot(111)

ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
ax.spines['bottom'].set_position('center')
ax.spines['left'].set_position('center')

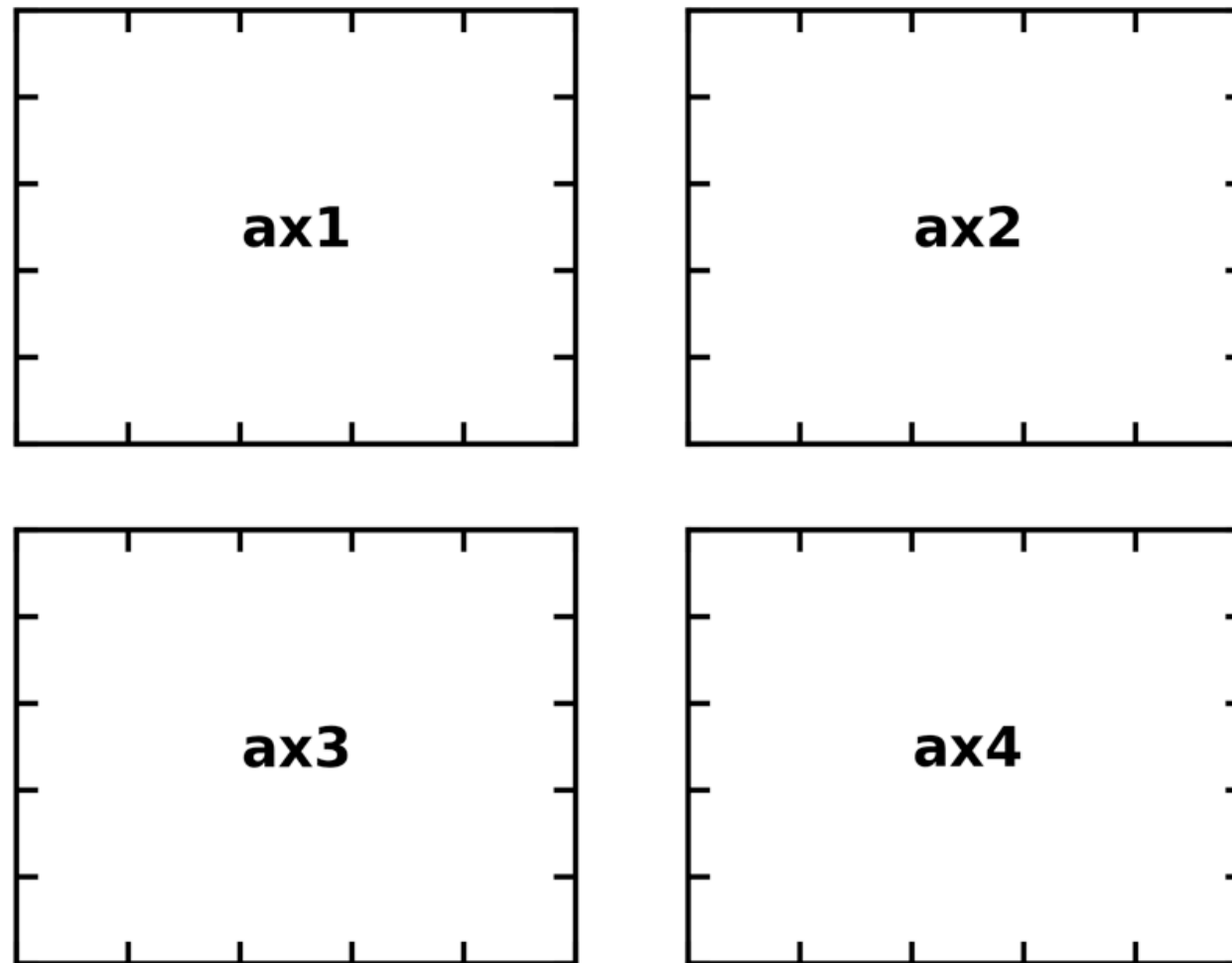
for tick in ax.xaxis.get_major_ticks():
    tick.tick2on = False

for tick in ax.yaxis.get_major_ticks():
    tick.tick2on = False

plt.show()
```

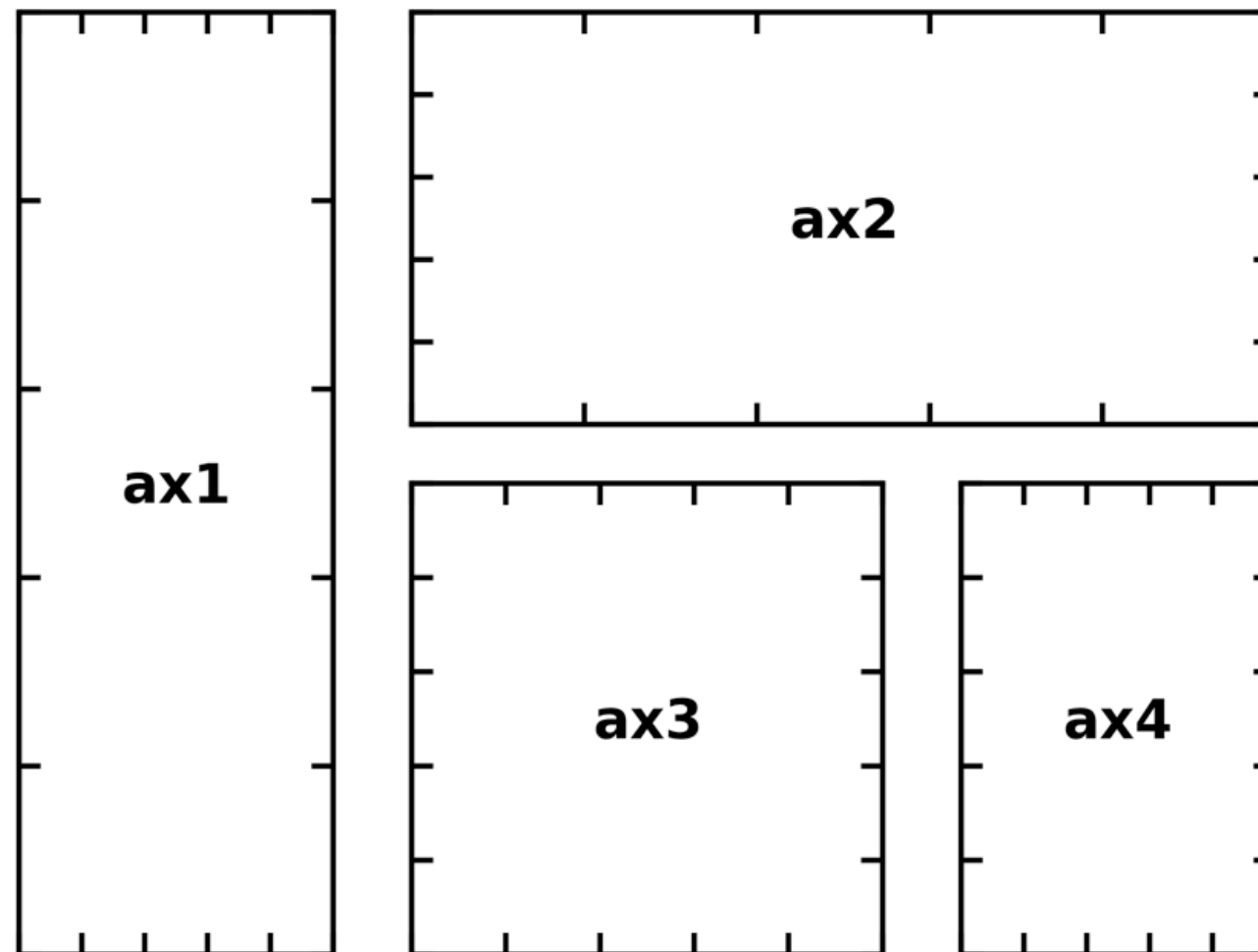


# Customizing location of Axes



```
ax1 = fig.add_subplot(221)
ax2 = fig.add_subplot(222)
ax3 = fig.add_subplot(223)
ax4 = fig.add_subplot(224)
```

# Customizing location of Axes



```
# add_axes((left, bottom, width, height))
```

```
ax1 = fig.add_axes((0.1, 0.1, 0.2, 0.8))
```

```
ax2 = fig.add_axes((0.35, 0.55, 0.55, 0.35))
```

```
ax3 = fig.add_axes((0.35, 0.1, 0.3, 0.4))
```

```
ax4 = fig.add_axes((0.7, 0.1, 0.2, 0.4))
```



**FREE!**

The memory required for a figure is not completely released until the figure is explicitly closed with `close()`.

```
import os
import glob
import matplotlib.pyplot as plt

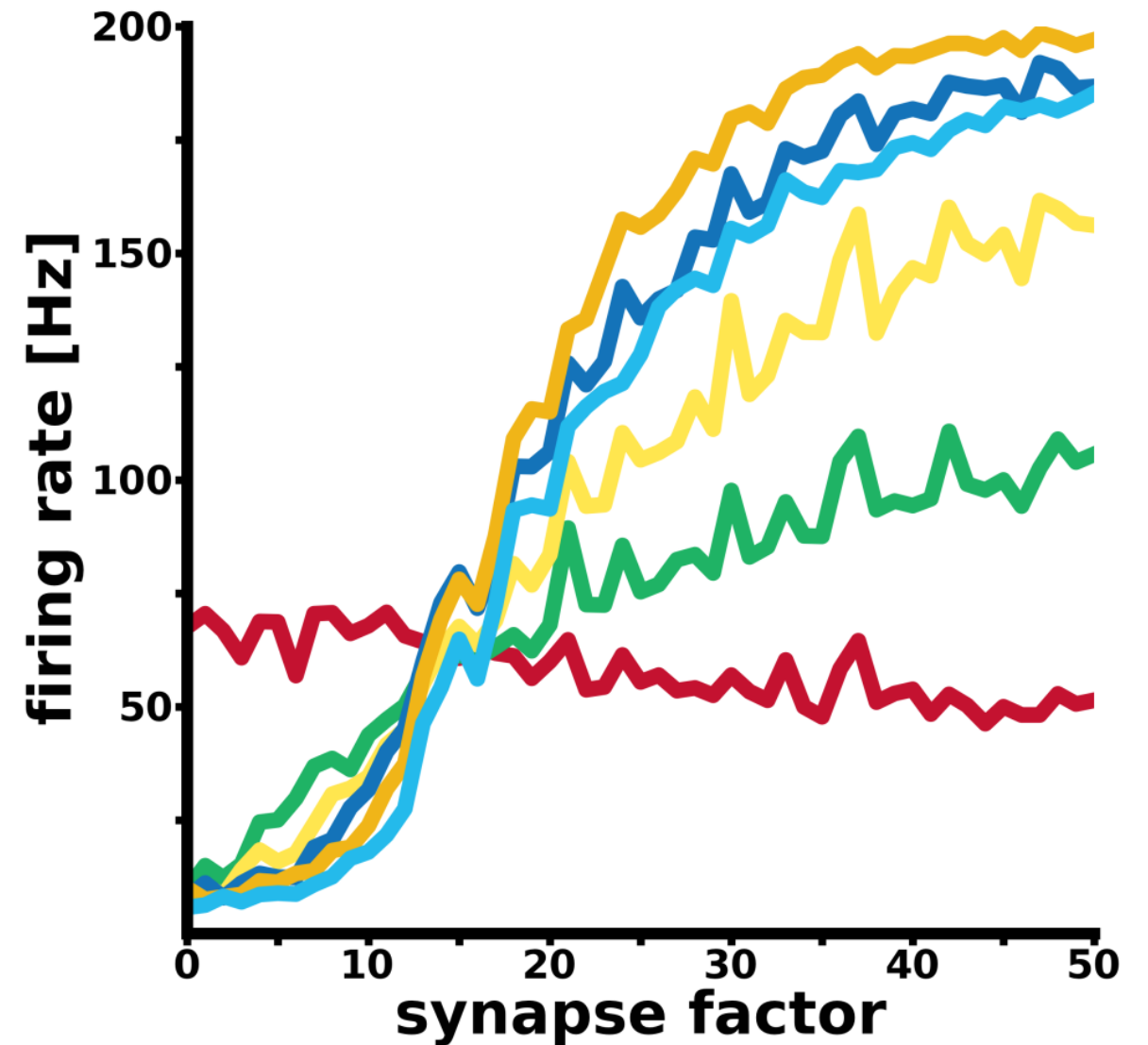
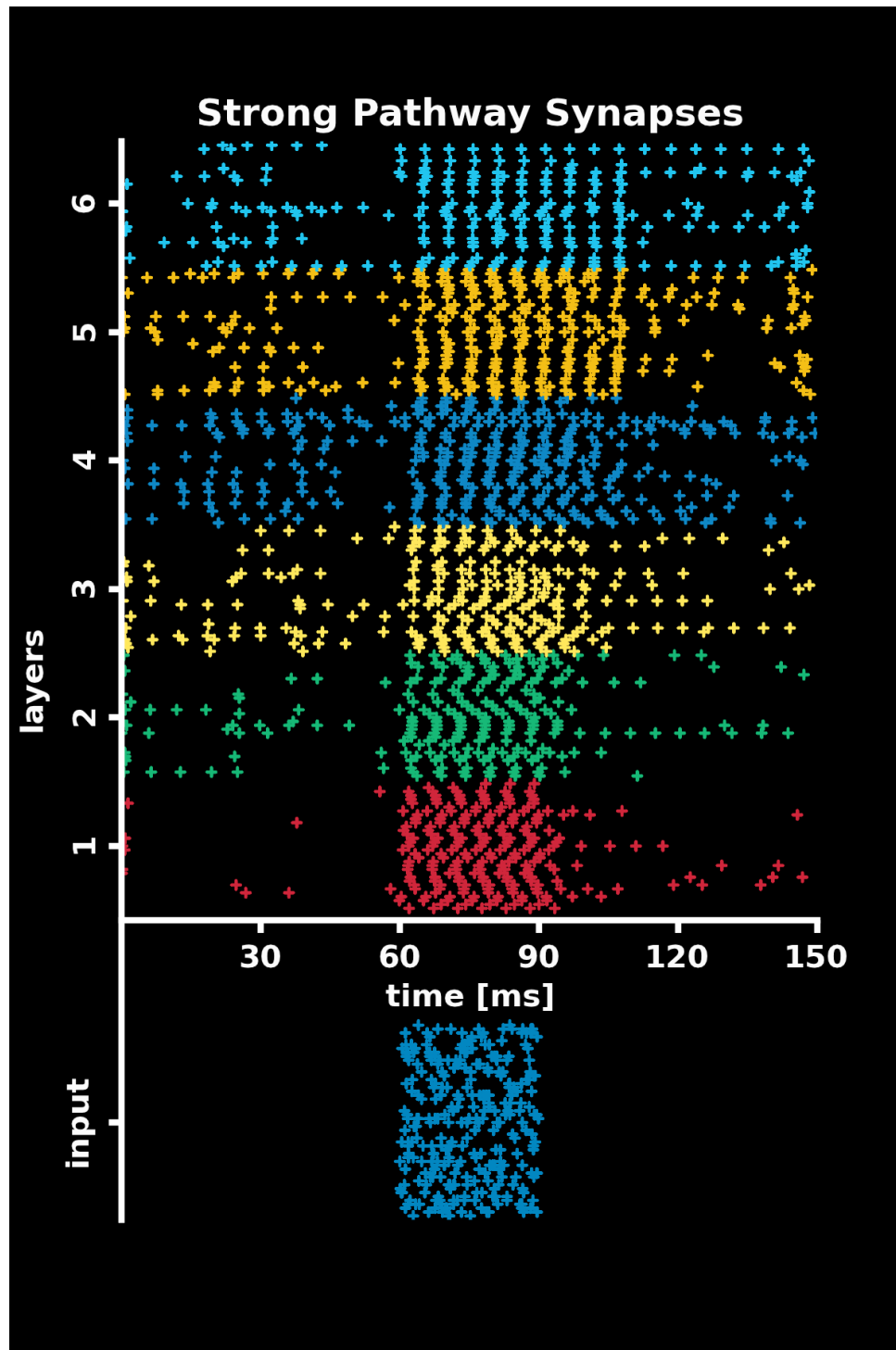
filelist = glob.glob('*.txt')
for fname in filelist:

    ...
    ...
    ...

    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot(x, y)
    plt.savefig(os.path.splitext(fname)[0])
    plt.close(fig)
```

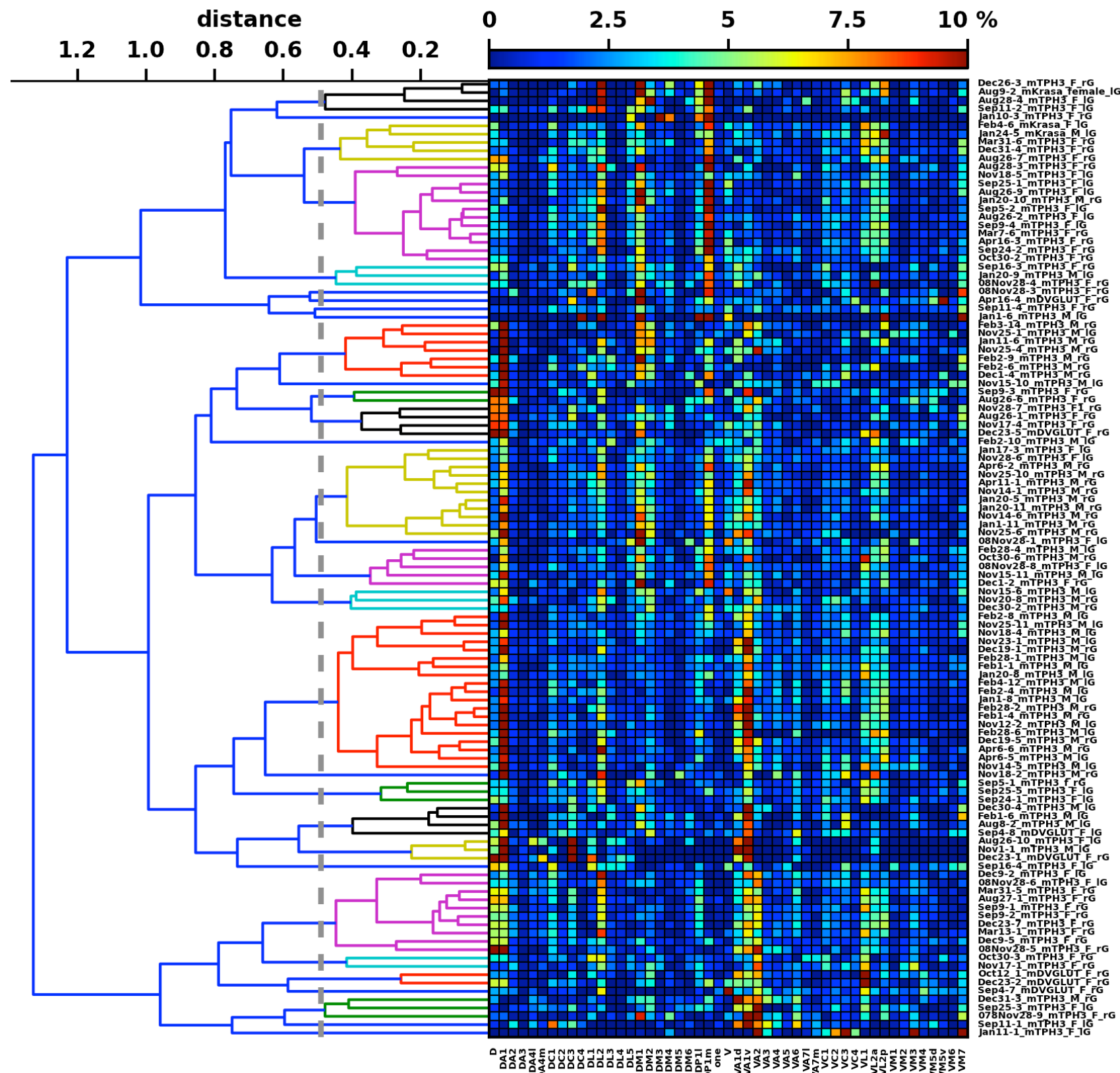


# Signal propagation

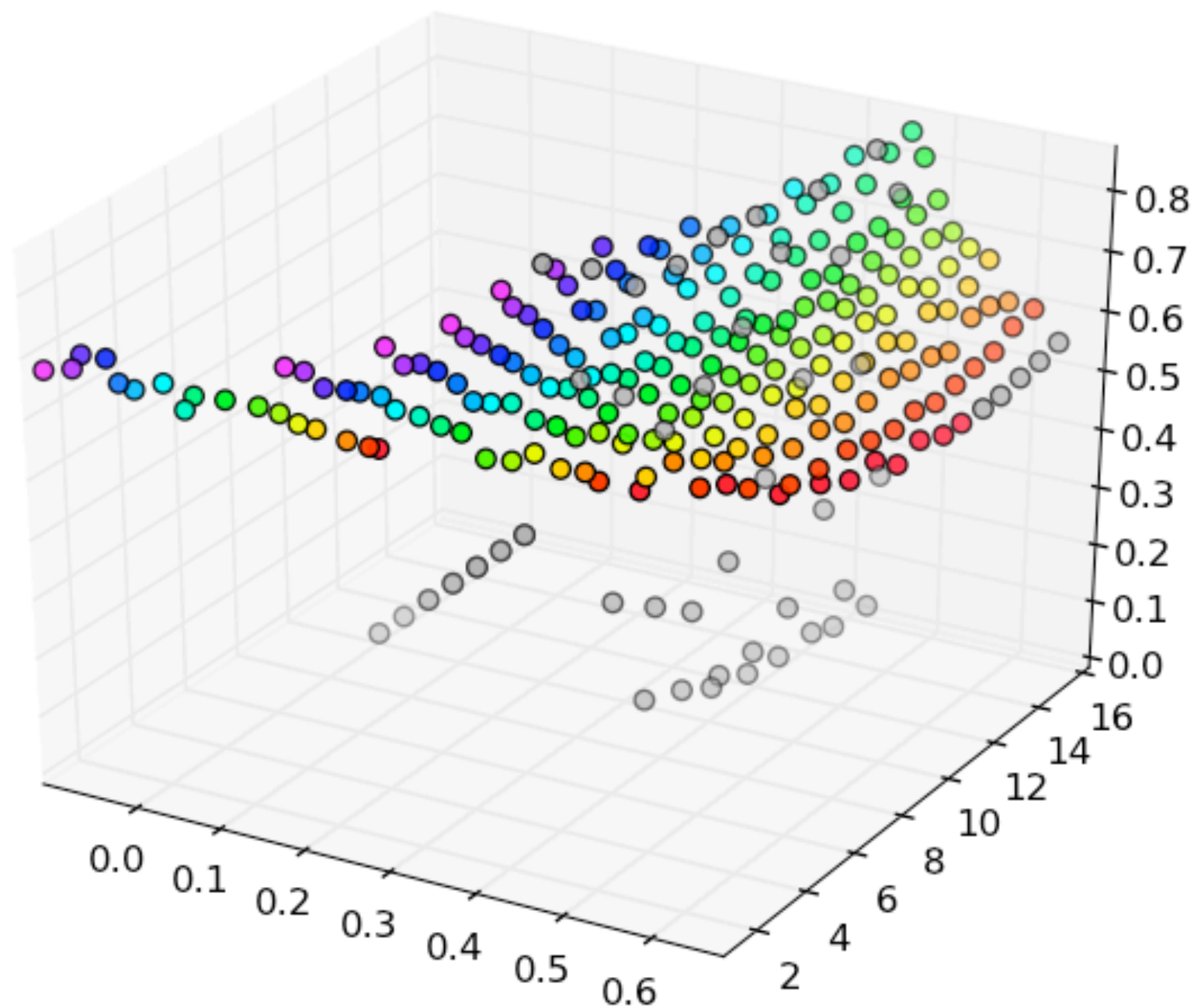


- Vogels TP, Abbott LF (2005) Signal propagation and logic gating in networks of integrate-and-fire neurons. J Neurosci 25: 10786-10795.
- Brian: a simulator for spiking neural networks in Python (<http://briansimulator.org>)

# Hierarchical clustering



`mpl_toolkits.mplot3d`  
provides some basic 3D plotting tools



May the Matplotlib be with You :)