

" USE THE MATPLOTLIB, LUKE "

WEN-WEI LIAO
gattacaliao@gmail.com

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



NO!

John Hunter

“

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

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

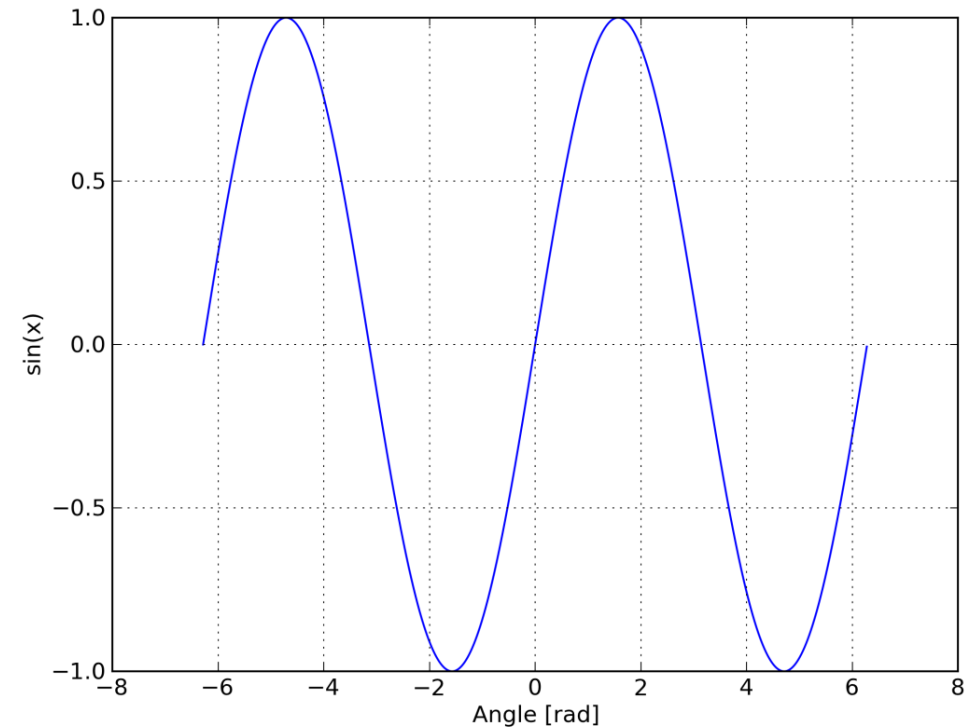
x = np.arange(0, 10, 0.1)
y = np.sin(x)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(x, y)
plt.show()
```

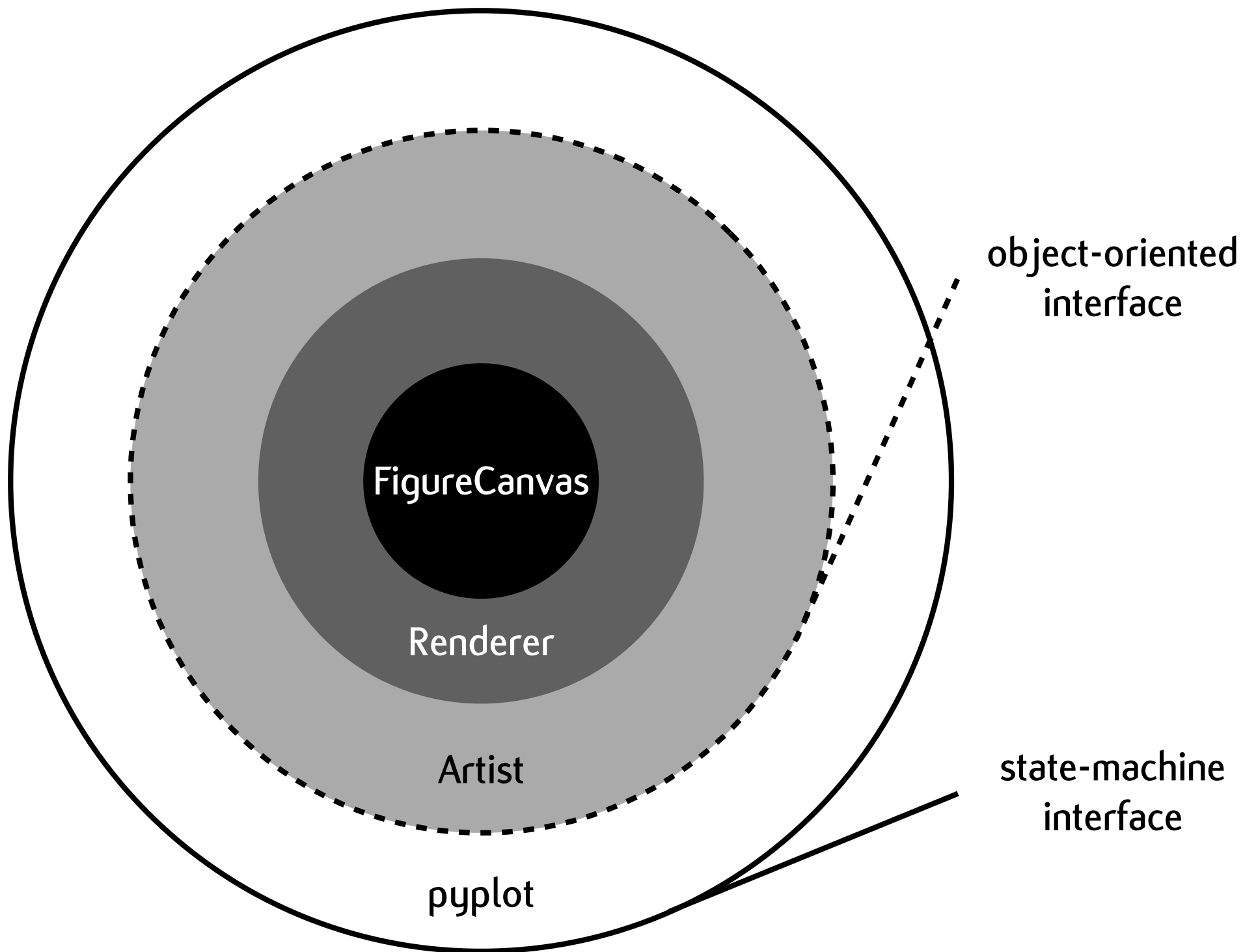
Simple example

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

x = np.arange(-2*np.pi, 2*np.pi, 0.01)
y = np.sin(x)

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(x, y)
ax.set_xlabel('Angle [rad]')
ax.set_ylabel('sin(x)')
ax.grid()
plt.show()
```





2 types of Artists

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

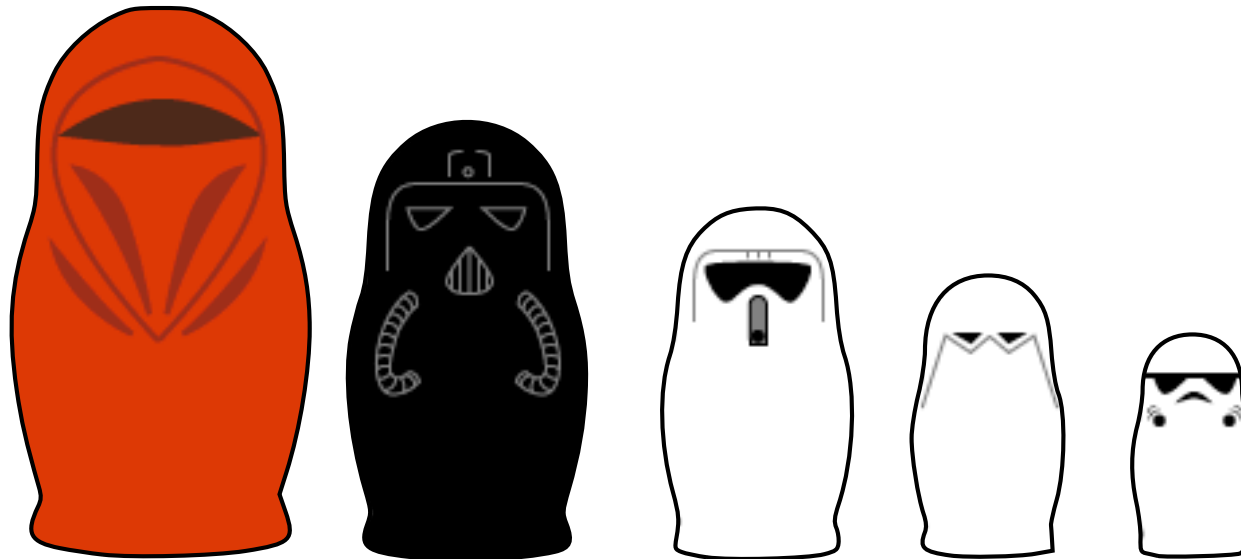
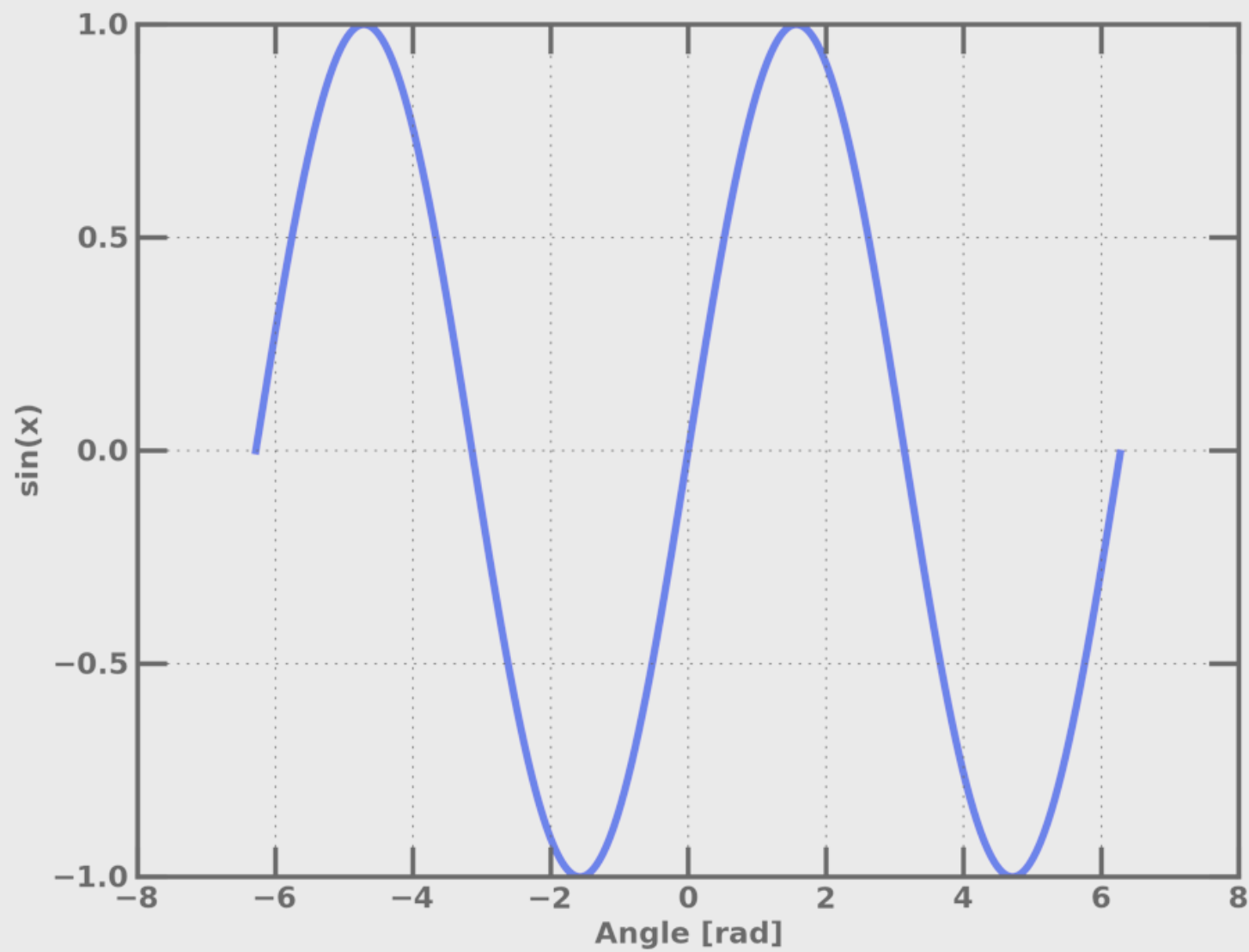


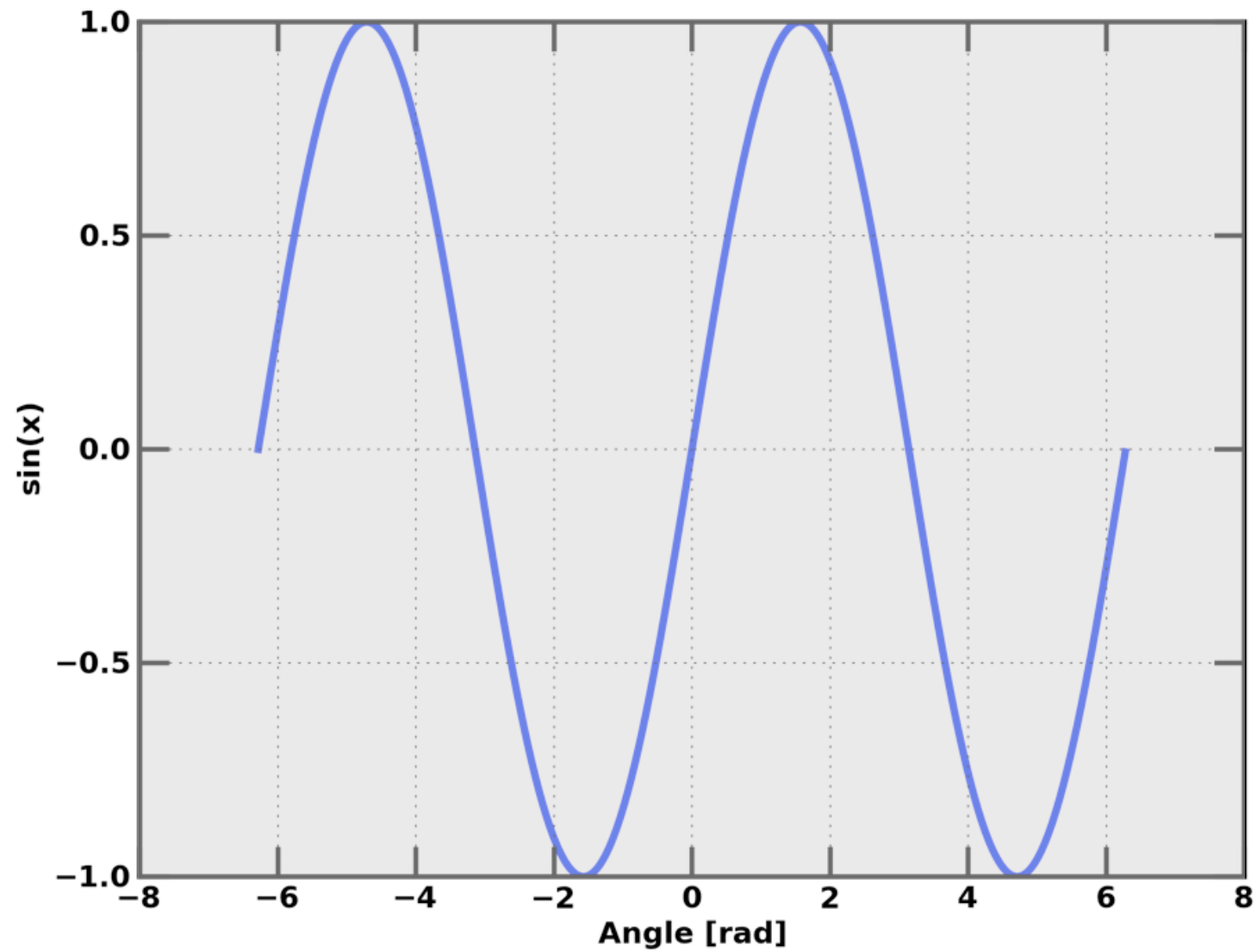
Figure Container

(matplotlib.figure.Figure)



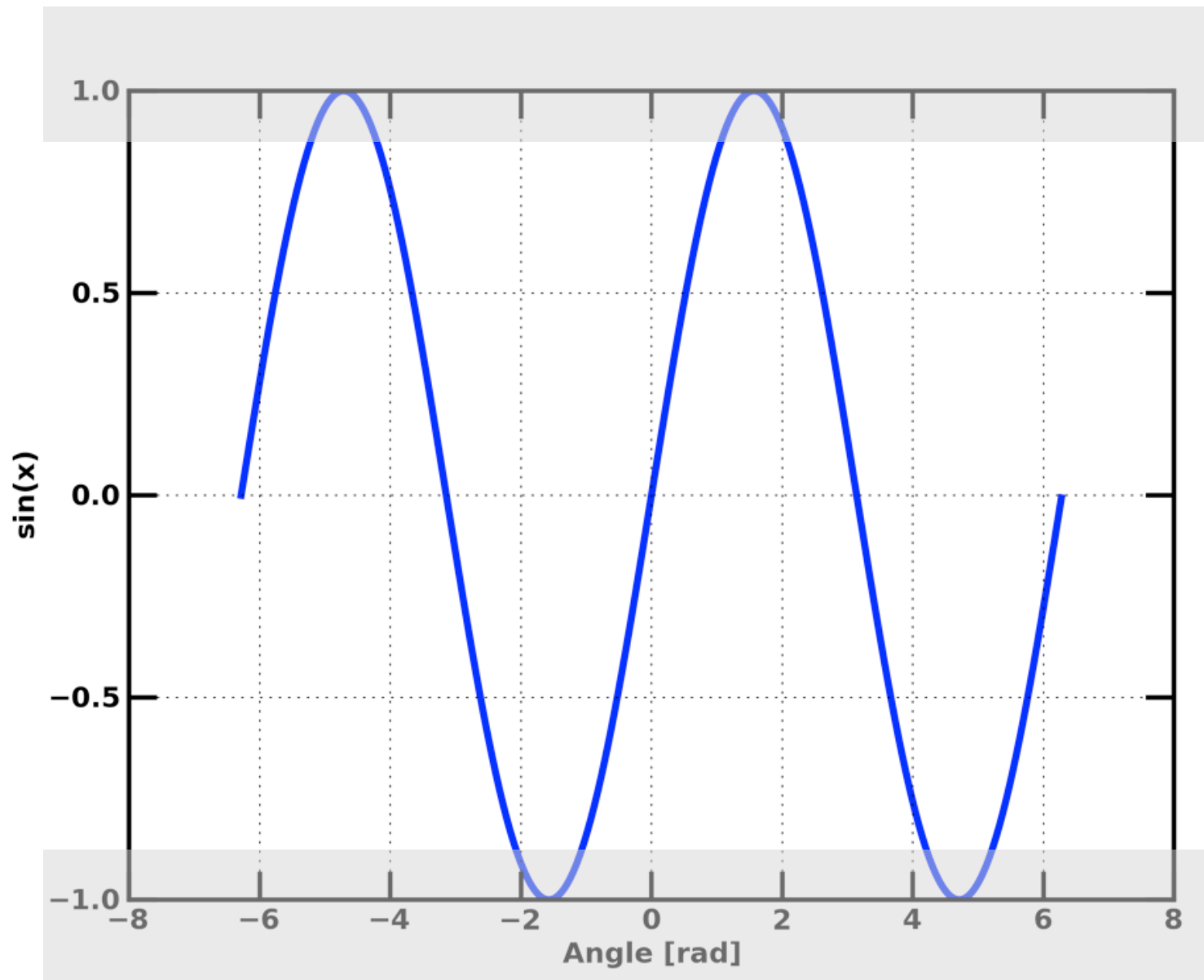
Axes Container

(matplotlib.axes.Axes)



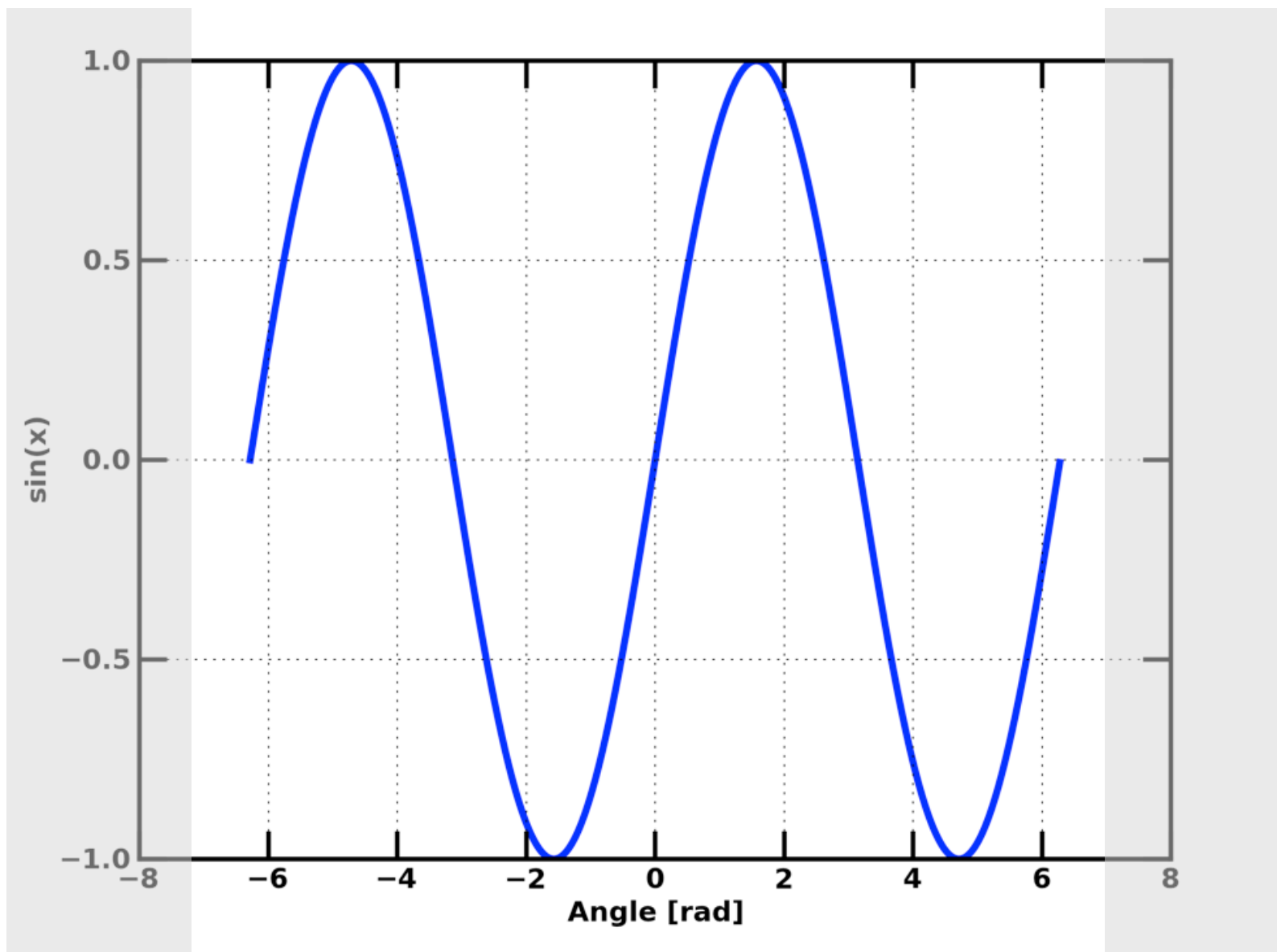
XAxis Container

(matplotlib.axis.Axis)



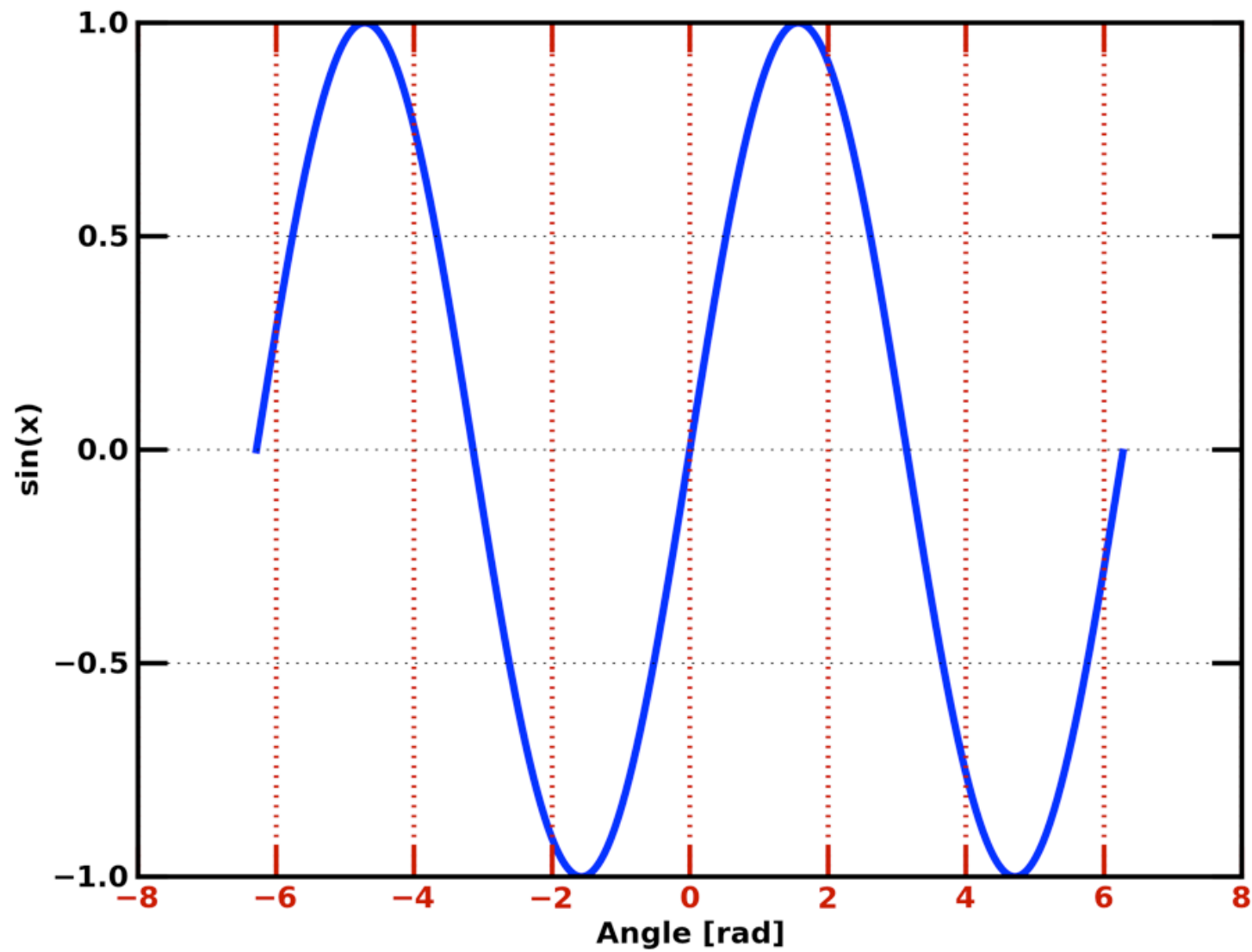
YAxis Container

(matplotlib.axis.Axis)



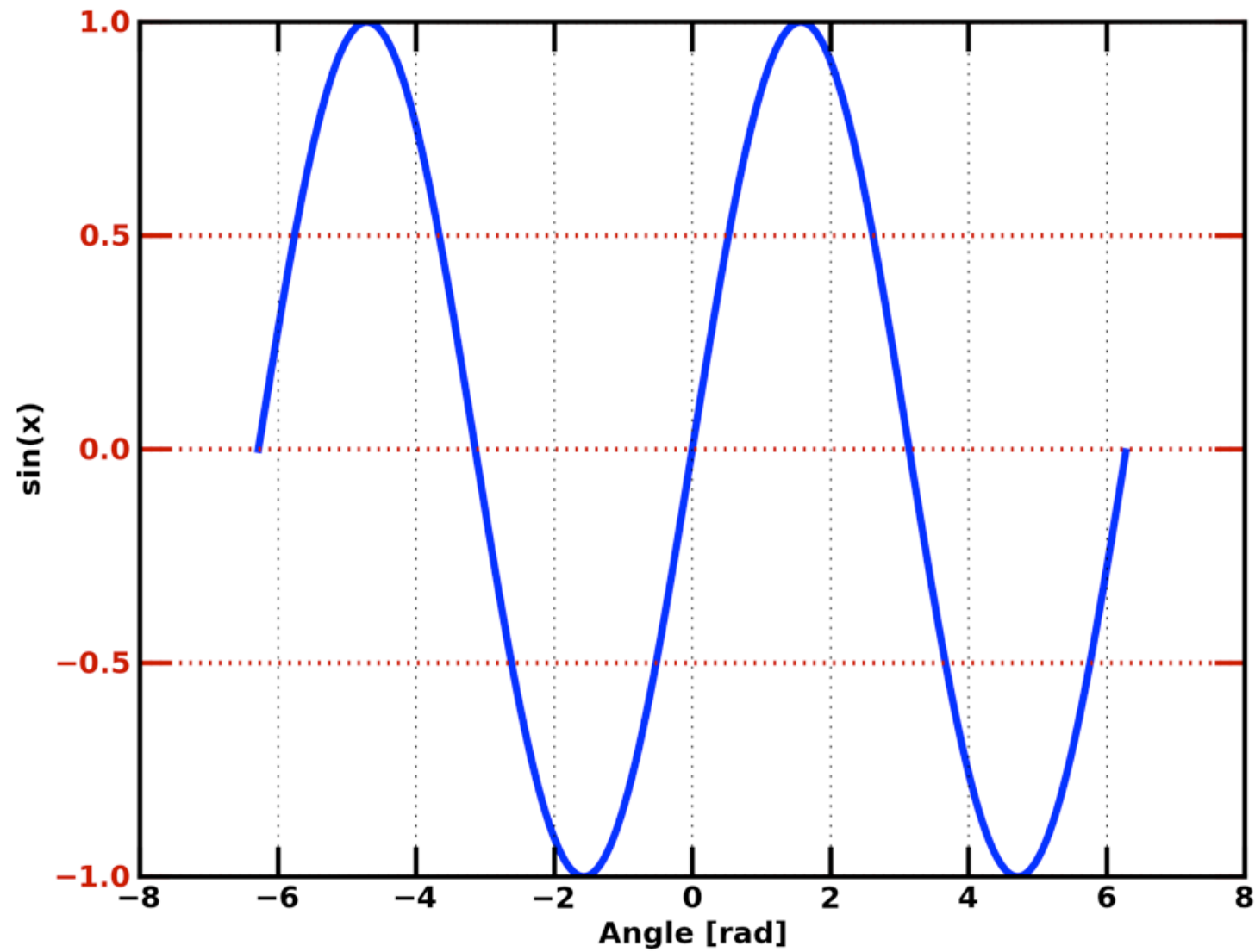
XTick Container

(matplotlib.axis.Tick)



YTick Container

(matplotlib.axis.Tick)



Customizing your objects

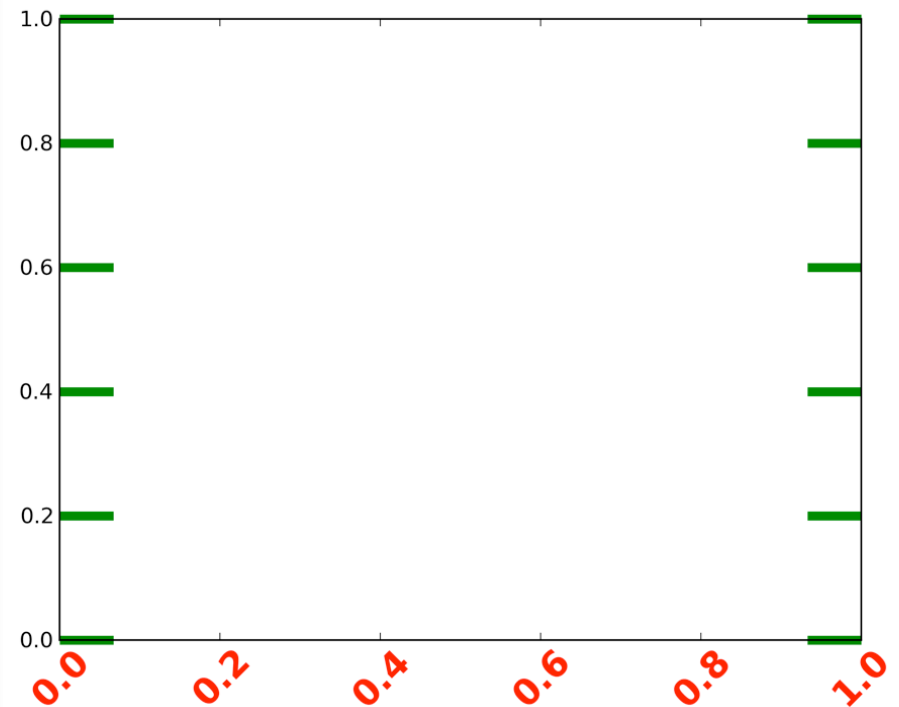
```
import matplotlib.pyplot as plt

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()
```



Customizing your objects

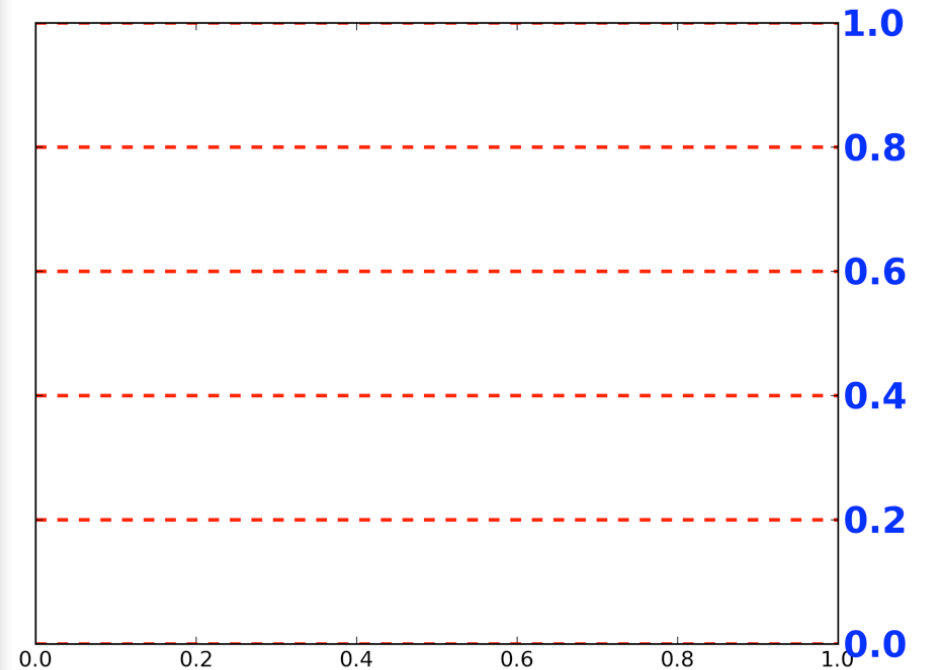
```
import matplotlib.pyplot as plt

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()
```

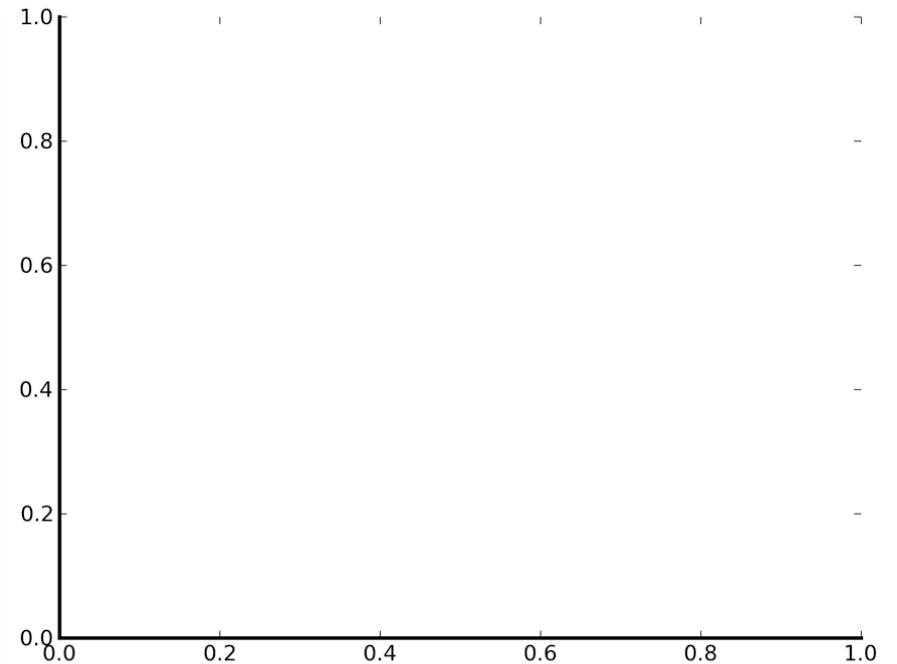


Customizing your objects

```
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_linewidth(2)
ax.spines['left'].set_linewidth(2)

plt.show()
```



Find all objects in a figure of a certain type

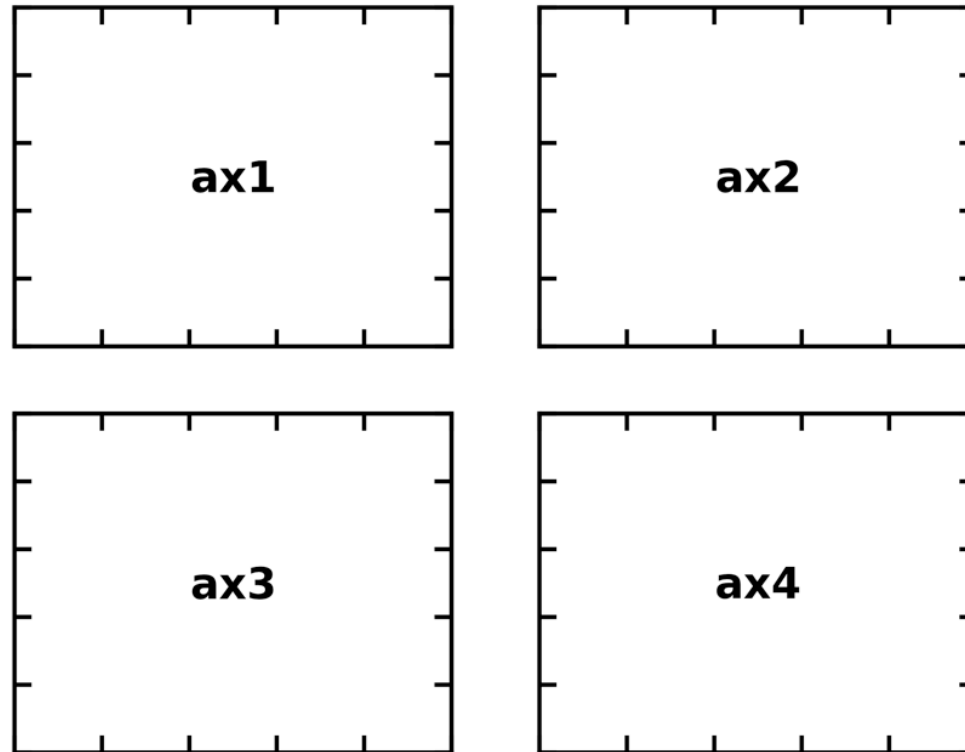
Find every object in the figure which has a `set_color` property and makes the object blue

```
def myfunc(x):  
    return hasattr(x, 'set_color')  
  
for o in fig.findobj(myfunc):  
    o.set_color('blue')
```

Filter on class instances

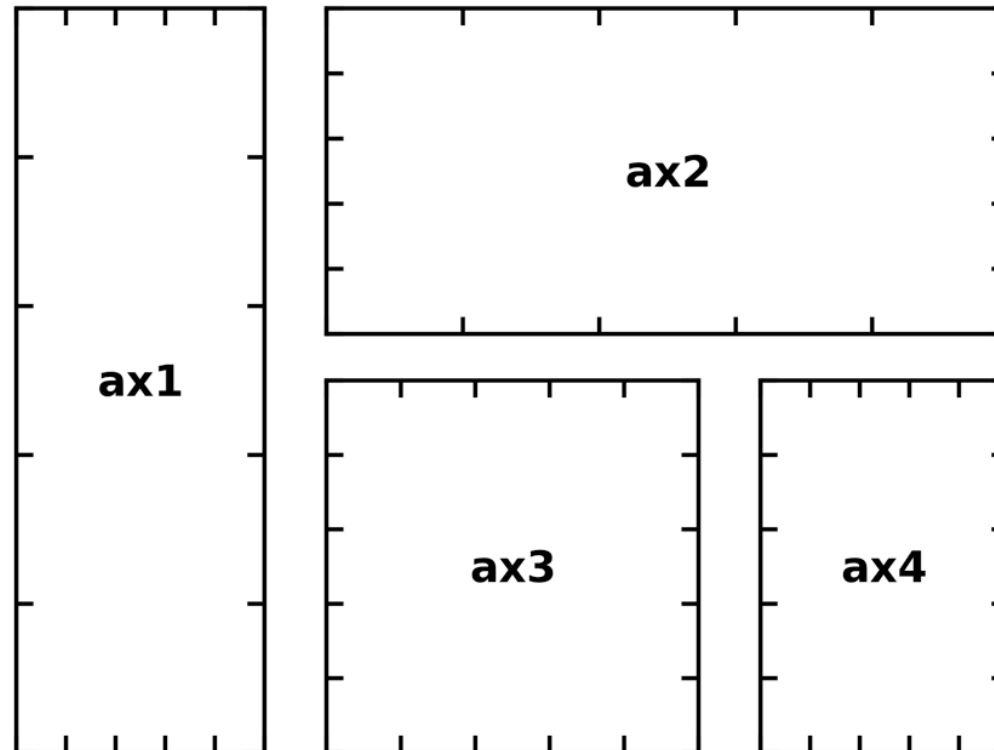
```
import matplotlib.text as text  
  
for o in fig.findobj(text.Text):  
    o.set_fontstyle('italic')
```

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))
```

A close-up, black and white comic book illustration of a man's face. He has a determined, shouting expression with his mouth wide open, showing teeth. His eyes are squeezed shut, and several sweat drops are flying off his forehead and cheeks. The word "FREE!" is written in large, bold, black, sans-serif capital letters across the center of his face. The background is white, and the man's skin is rendered in shades of gray with black outlines for his features.

FREE!

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

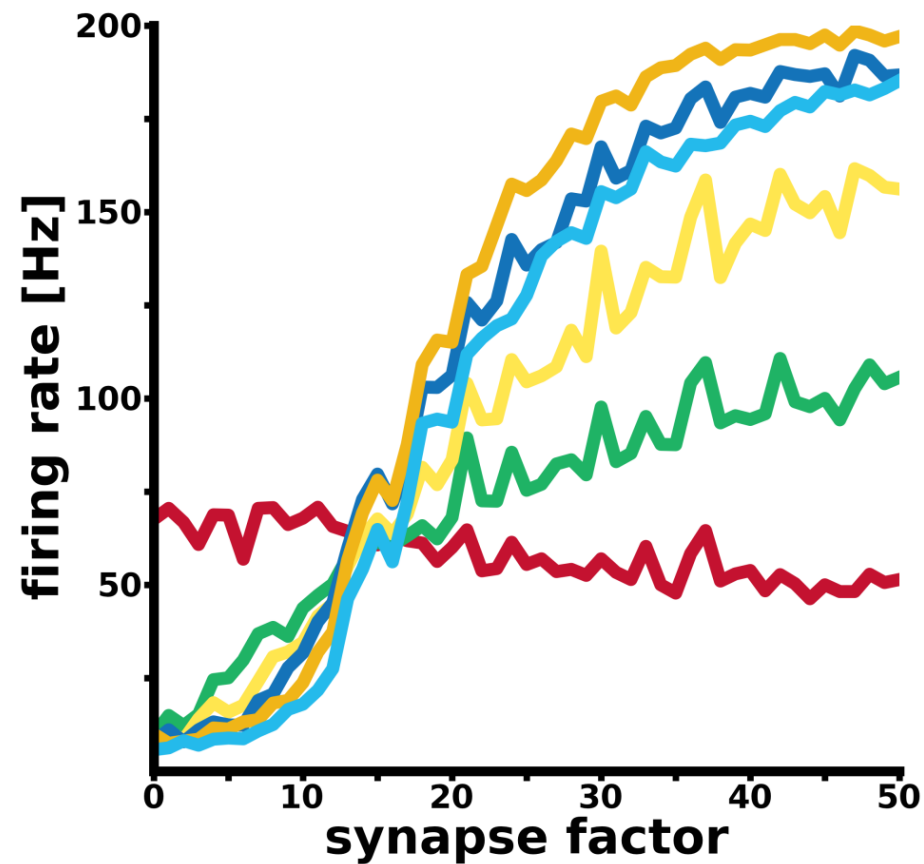
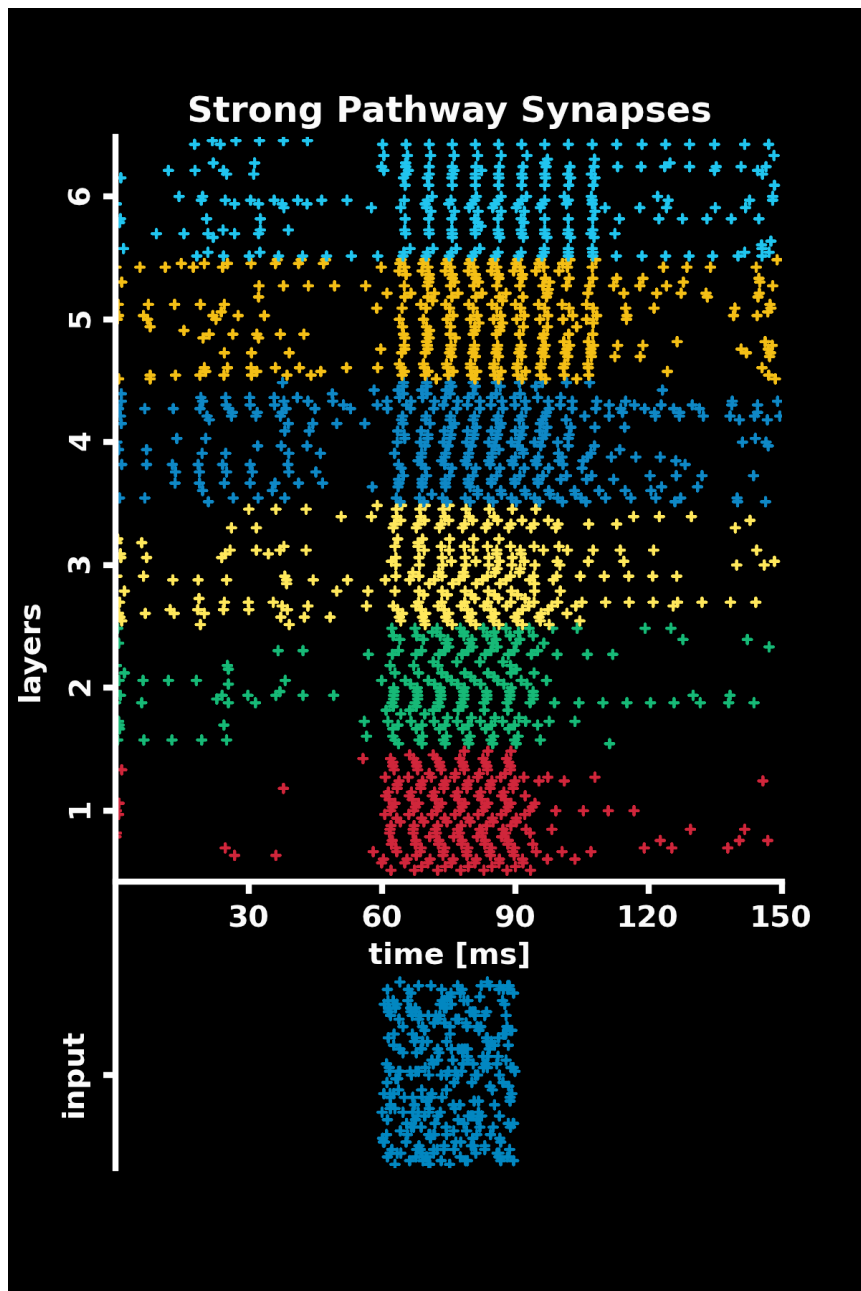
```
import os
import glob
import matplotlib.pyplot as plt

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

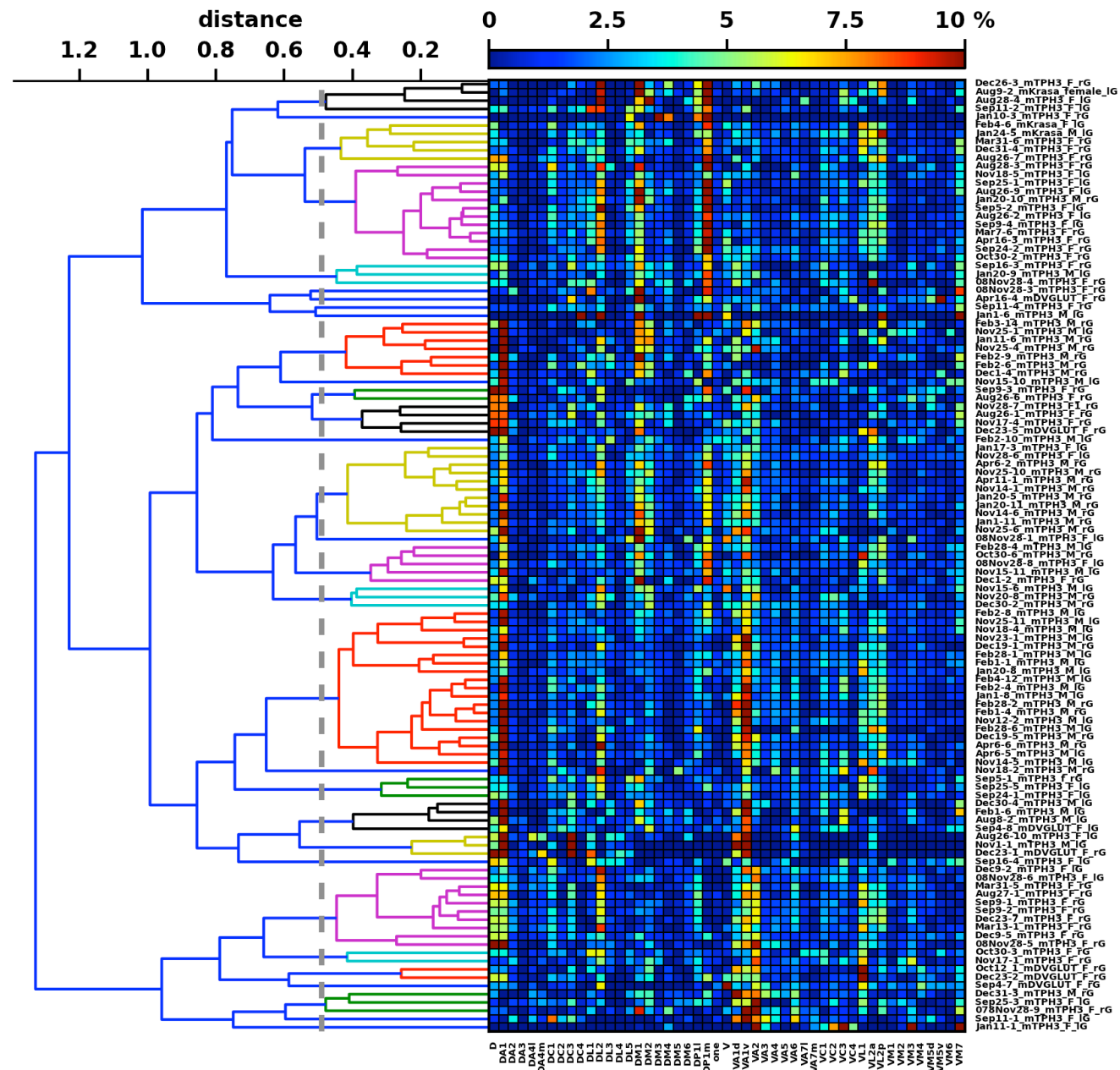
    ...
    ...
    ...

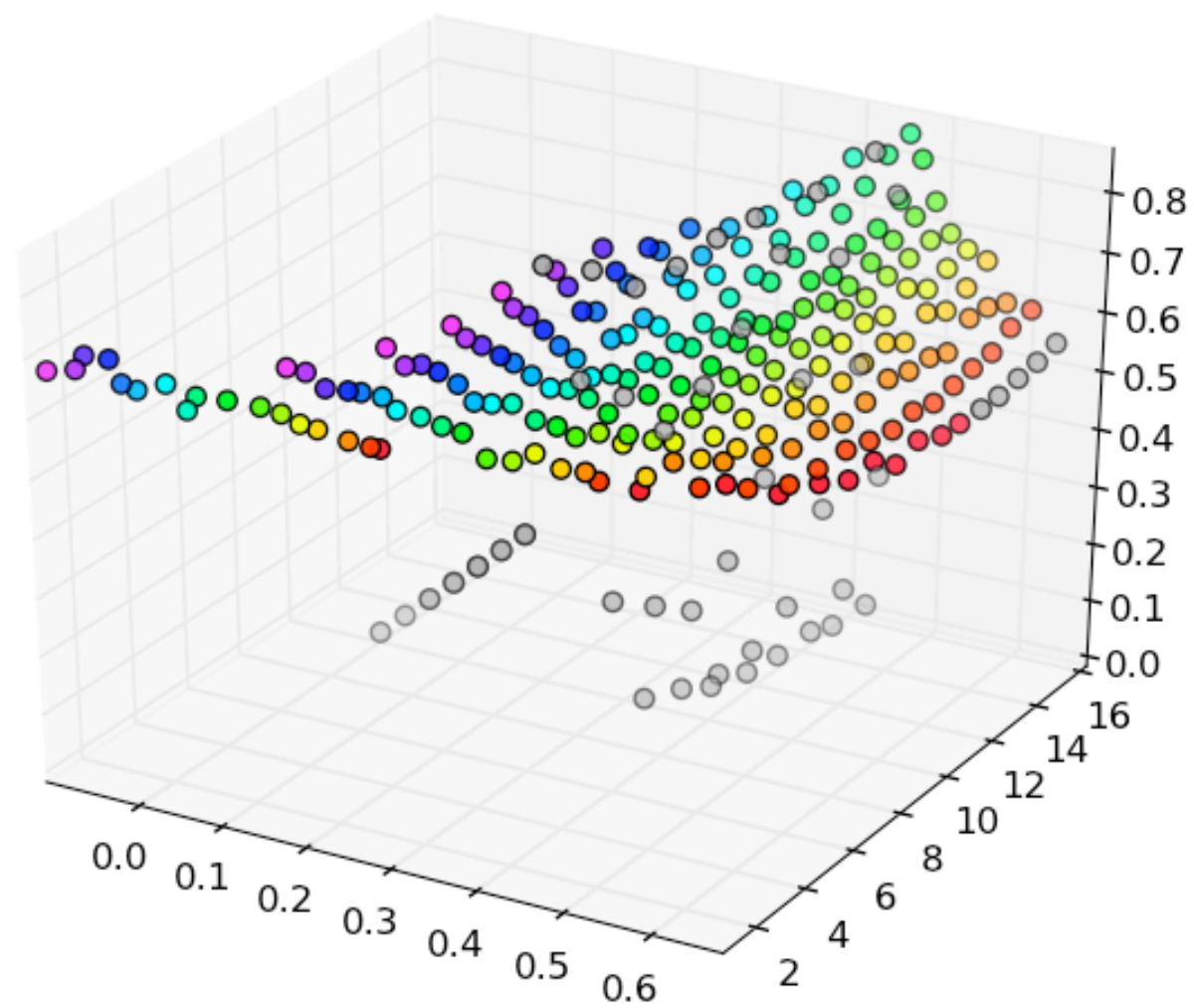
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot(x, y)
    plt.savefig(os.path.splitext(fname)[0])
    plt.close(fig)
```


Signal propagation



Hierarchical clustering





May the Matplotlib be with You :)