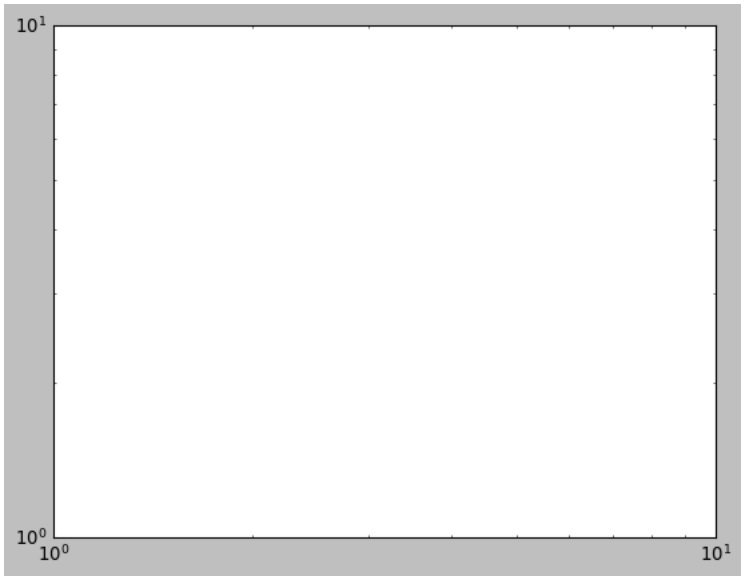
In [2]:
```
import matplotlib.pyplot as plt
plt.style.use("classic")
%matplotlib inline
import numpy as np
```

In [4]:
```
ax=plt.axes(xscale='log', yscale='log')
ax.grid();
```



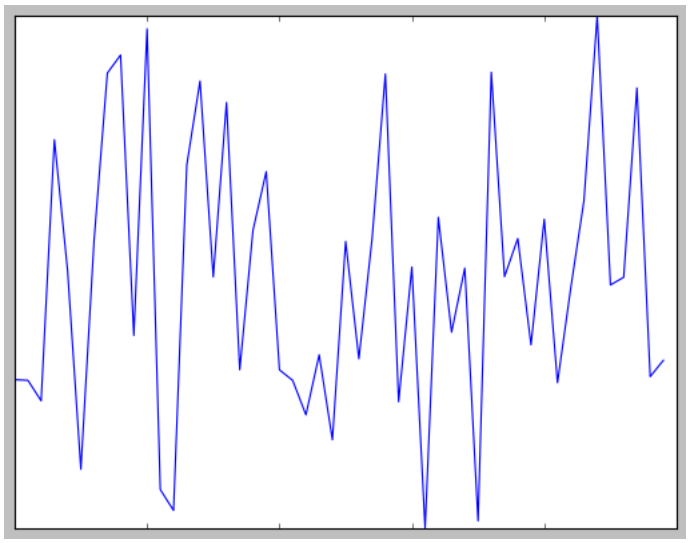
In [5]:
```
#to get location oject of major and minor ticks we have to use locator
print(ax.xaxis.get_major_locator())#major ticks locationn object
print(ax.xaxis.get_minor_locator())#minnor ticks location object
```

```
<matplotlib.ticker.LogLocator object at 0x00000257CAD563A0>
<matplotlib.ticker.LogLocator object at 0x00000257CAE19DF0>
```

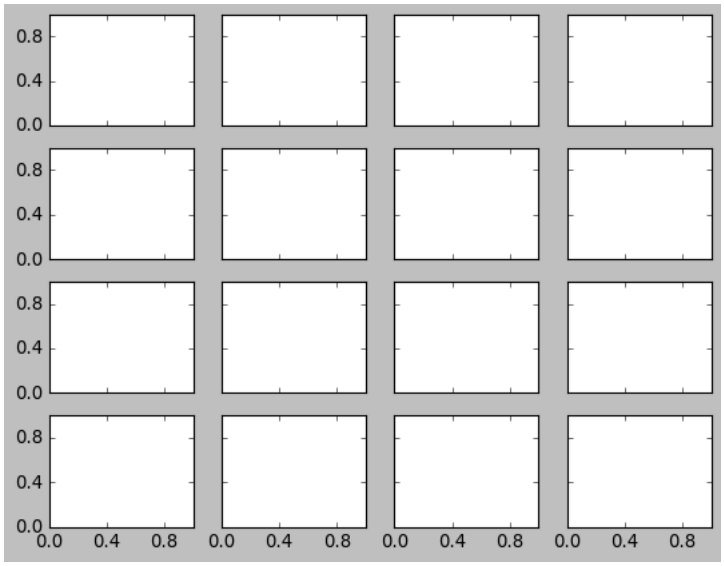
In [6]:
```
#formatter object
print(ax.xaxis.get_major_formatter())
print(ax.xaxis.get_minor_formatter())
```

```
<matplotlib.ticker.LogFormatterSciNotation object at 0x00000257CAD46D00>
<matplotlib.ticker.LogFormatterSciNotation object at 0x00000257CAE19FD0>
```

In [11]:
```
#hiding ticks and lbels using NullLocator and NullFormatter and set
ax=plt.axes()
ax.plot(np.random.rand(50))
ax.xaxis.set_major_formatter(plt.NullFormatter())#removes labels in x axis
ax.yaxis.set_major_locator(plt.NullLocator())#removes major ticks in y axis
```

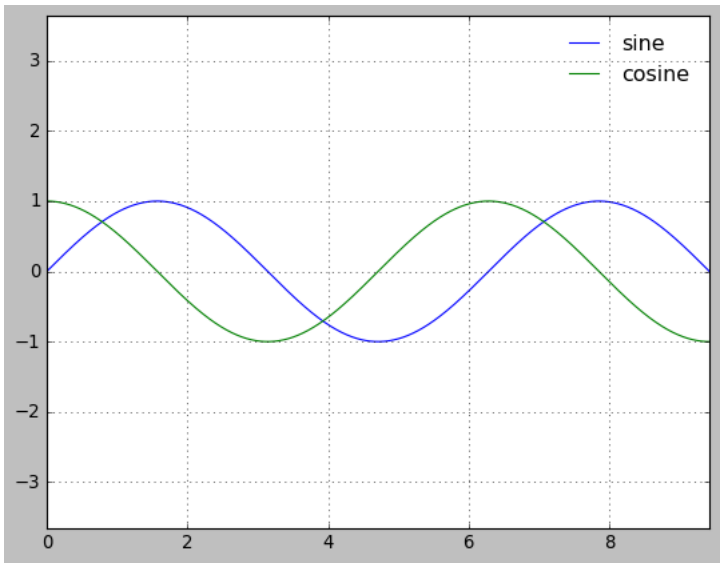


In [19]:
```
fig,ax=plt.subplots(4,4,sharex=True,sharey=True)
#to specify maximum number of ticks displayed we use plt.MaxNLocator()
for axi in ax.flat:
    axi.xaxis.set_major_locator(plt.MaxNLocator(3))
    axi.yaxis.set_major_locator(plt.MaxNLocator(3))
```



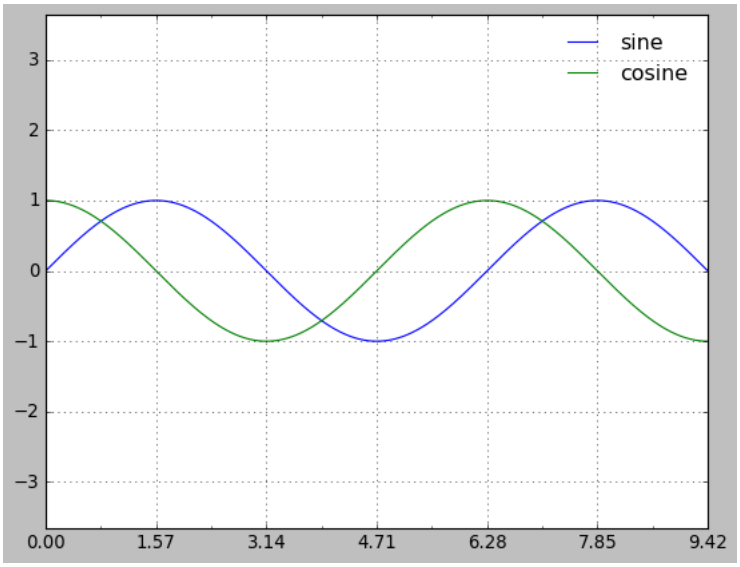
In [21]:
```
#fancy ticks
fig,ax=plt.subplots()
x=np.linspace(0,np.pi*3,1000)
ax.plot(x,np.sin(x),label="sine")
ax.plot(x,np.cos(x),label="cosine")
ax.grid(True)#setting grid
ax.legend(frameon=False)
ax.axis("equal")
ax.set_xlim(0,3*np.pi)
```

Out[21]: (0.0, 9.42477796076938)



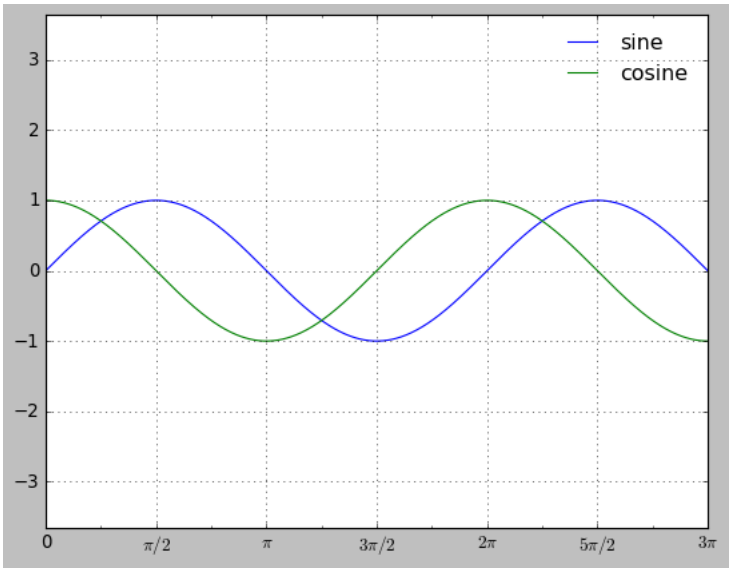
In [25]:
```
#MultipleLocator() is used to place ticks at positions at multiples of particular number
ax.xaxis.set_major_locator(plt.MultipleLocator((np.pi)/2))
ax.xaxis.set_minor_locator(plt.MultipleLocator((np.pi)/4))
fig
```

Out[25]:



In [27]:
```
def format_func(value, tick_number):
    # find number of multiples of pi/2
    N = int(np.round(2 * value / np.pi))
    if N == 0:
        return "0"
    elif N == 1:
        return r"$\pi/2$"
    elif N == 2:
        return r"$\pi$"
    elif N % 2 > 0:
        return r"${0}\pi/2$".format(N)
    else:
        return r"${0}\pi$".format(N // 2)
#function formatter is used to set ticks with help of user defined fucntion
ax.xaxis.set_major_formatter(plt.FuncFormatter(format_func))
fig
```

Out[27]:



In [28]:
```
#LiearLocator-evenly placed ticks from min to max
#LogLocator-Logarithmly ticks min and max
#AutoLocator-MaxNLocator with simple defaults
#AutoMinorLocator-Locator for Minor ticks
#IndexFormatter-Set strings from list of labels
#FormatStrFormatter-Use format string for each value
```