

## INDICE

- [Subplots: griglie regolari di grafici](#)
  - [2 grafici in colonna: subplot\(2,1,\\*\)](#)
  - [2 grafici in riga: subplot\(1,2,\\*\)](#)
    - [4 grafici in 2 righe e due colonne : subplot\(2,2,\\*\)](#)
- [Griglie meno regolari con Gridspec](#)
- [Disposizione libera degli axes](#)
  - [uno dentro l'altro](#)
  - [uno dentro l'altro v2](#)

```
In [1]: import matplotlib
import matplotlib.pyplot as plt
import numpy as np
```

[http://sage.math.washington.edu/home/wstein/www/home/agc/lit/python/mpl\\_brain\\_surg.pdf](http://sage.math.washington.edu/home/wstein/www/home/agc/lit/python/mpl_brain_surg.pdf)

## Subplots: griglie regolari di grafici

con il comando [pyplot.subplot\(\)](#) posso creare una griglia regolare di grafici

### 2 grafici in colonna: subplot(2,1,\*)

creo questo grafico



```
In [3]: f=plt.figure(figsize=(6, 4))

#subplot(2, 1, 1)
ax1=plt.subplot(2, 1, 1)
#accedo ai ticks direttamente dall axes
ax1.set_xticklabels(())
#oppure accedo ai ticks attraverso l'oggetto xaxis o yaxis
ax1.yaxis.set_ticklabels(())
ax1.text(0.5, 0.5, 'subplot(2,1,1)', ha='center', va='center',
        size=24, alpha=.5)
```

```

#subplot(2, 1, 2)
ax2=pl.subplot(2, 1, 2)
ax2.set_xticklabels(())
ax2.set_yticklabels(())
ax2.text(0.5, 0.5, 'subplot(2,1,2)', ha='center', va='center',
        size=24, alpha=.5)

pl.savefig("./img/matplotlib20.png")
pl.show()

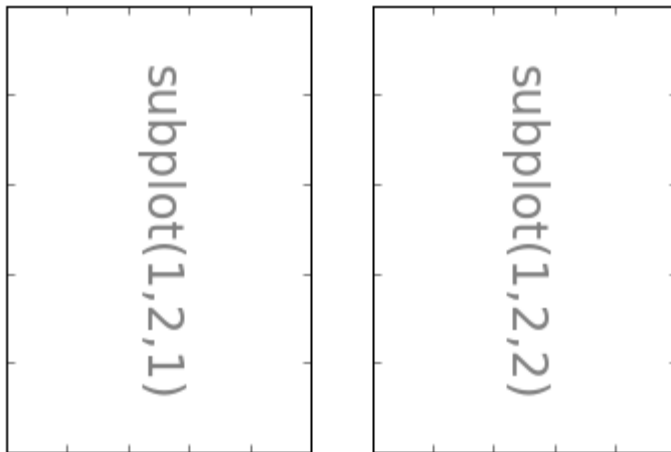
```



## 2 grafici in riga: subplot(1,2,\*)

[http://matplotlib.org/api/pyplot\\_api.html#matplotlib.pyplot.text](http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.text)

creo questo grafico



```

In [4]: f = pl.figure(figsize(6,4))
a1 = pl.subplot(1,2,1)
a2 = pl.subplot(1,2,2)

a1.set_xticklabels(()),a1.set_yticklabels(())
a2.set_xticklabels(()),a2.set_yticklabels(())

a1.text(0.5, 0.5, 'subplot(1,2,1)', ha='center', va='center', rotation=-90,
        size=24, alpha=.5)
a2.text(0.5, 0.5, 'subplot(1,2,2)', ha='center', va='center', rotation=-90,
        size=24, alpha=.5)

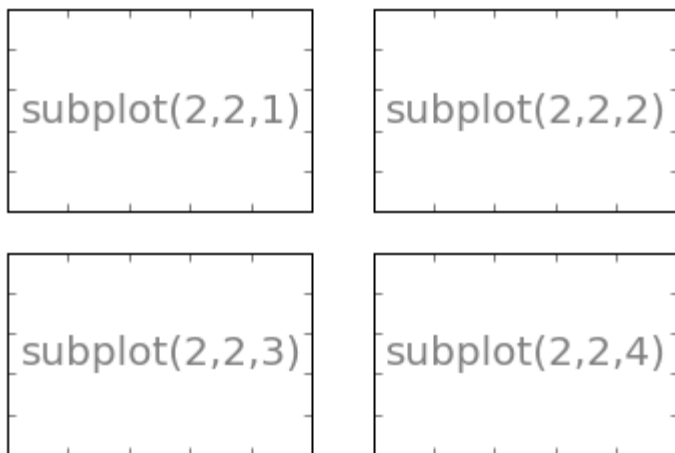
pl.savefig("./img/matplotlib21.png")
pl.show()

```



## 4 grafici in 2 righe e due colonne : subplot(2,2,\*)

creo questo grafico



```
In [5]: pl.figure(figsize=(6, 4))

#subplot(2, 2, 1)
a1=pl.subplot(2, 2, 1)
a1.set_xticklabels(())
a1.set_yticklabels(())
a1.text(0.5, 0.5, 'subplot(2,2,1)', ha='center', va='center',
        size=20, alpha=.5)

#subplot(2, 2, 2)
a2=pl.subplot(2, 2, 2)
a2.set_xticklabels(())
a2.set_yticklabels(())
a2.text(0.5, 0.5, 'subplot(2,2,2)', ha='center', va='center',
        size=20, alpha=.5)

#subplot(2, 2, 3)
a3=pl.subplot(2, 2, 3)
a3.set_xticklabels(())
a3.set_yticklabels(())

a3.text(0.5, 0.5, 'subplot(2,2,3)', ha='center', va='center',
        size=20, alpha=.5)

#subplot(2, 2, 3)
a4=pl.subplot(2, 2, 4)
a4.set_xticklabels(())
a4.set_yticklabels(())
a4.text(0.5, 0.5, 'subplot(2,2,4)', ha='center', va='center',
        size=20, alpha=.5)
pl.savefig("./img/matplotlib22.png")

pl.show()
```



Griglie meno regolari con Gridspec

creo questo grafico



```
In [6]: import matplotlib.gridspec as gridspec

pl.figure(figsize=(6, 4))
#definisco una matrice di base
G = gridspec.GridSpec(3, 3)

#specifico la disposizione dell'axes nella griglia in termini di elementi di G
axes_1 = pl.subplot(G[0, :])
axes_1.set_xticks(())
axes_1.set_yticks(())
axes_1.text(0.5, 0.5, 'Axes 1', ha='center', va='center', size=24, alpha=.5)

axes_2 = pl.subplot(G[1, :-1])
axes_2.set_xticks(())
axes_2.set_yticks(())
axes_2.text(0.5, 0.5, 'Axes 2', ha='center', va='center', size=24, alpha=.5)

axes_3 = pl.subplot(G[1:, -1])
axes_3.set_xticks(())
axes_3.set_yticks(())
axes_3.text(0.5, 0.5, 'Axes 3', ha='center', va='center', size=24, alpha=.5)

axes_4 = pl.subplot(G[-1, 0])
axes_4.set_xticks(())
axes_4.set_yticks(())
axes_4.text(0.5, 0.5, 'Axes 4', ha='center', va='center', size=24, alpha=.5)

axes_5 = pl.subplot(G[-1, -2])
axes_5.set_xticks(())
axes_5.set_yticks(())
axes_5.text(0.5, 0.5, 'Axes 5', ha='center', va='center', size=24, alpha=.5)

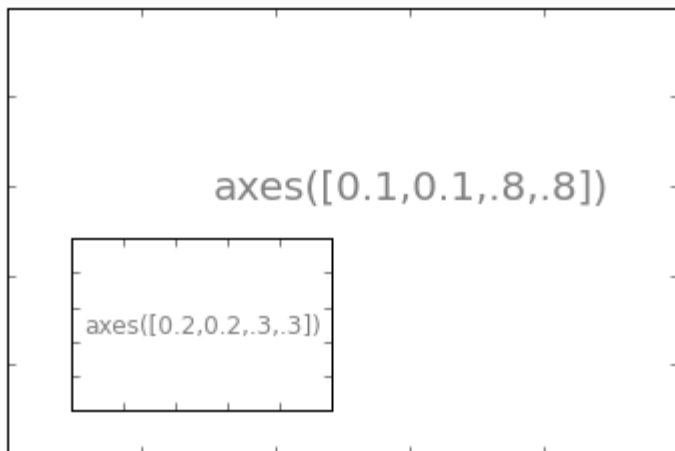
pl.savefig("./img/matplotlib23.png")
pl.show()
```



# Disposizione libera degli axes

uno dentro l'altro

creo questo grafico



```
In [9]: f=plt.figure()
f.patch.set_visible(True)
f.patch.set_color('y')
ax1 = plt.axes([.1, .1, .8, .8])
#ax1=plt.subplot(111)
ax1.set_xticklabels(())
ax1.set_yticklabels(())
ax1.text(.6, .6, 'axes([0.1,0.1,.8,.8])', ha='center', va='center',
        size=20, alpha=.5)

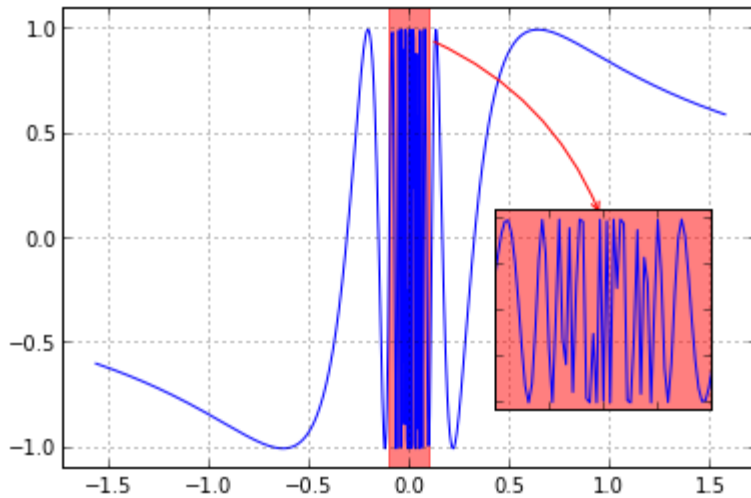
ax2 = plt.axes([.2, .2, .3, .3])
ax2.set_xticklabels(())
ax2.set_yticklabels(())
ax2.text(.5, .5, 'axes([0.2,0.2,.3,.3])', ha='center', va='center',
        size=12, alpha=.5)

plt.savefig("./img/matplotlib24.png")
plt.show()
```



uno dentro l'altro v2

creo questo grafico



```
In [11]: import numpy as np
import matplotlib.patches as patches
import matplotlib.transforms as transforms

x = np.linspace(-np.pi/2,np.pi/2,1000)
y = np.sin(1/x)

ax1=pl.axes([.1, .1, .8, .8])
ax1.set_xlim(min(x)*1.1,max(x)*1.1)
ax1.set_ylim(min(y)*1.1,max(y)*1.1)
ax1.grid()
ax1.plot(x,y)

ax2=pl.axes([.6, .2, .25, .35])
ax2.set_xlim(-0.1,0.1)
ax2.set_ylim(min(y)*1.1,max(y)*1.1)
ax2.set_xticklabels(())
ax2.set_yticklabels(())
ax2.plot(x,y)
ax2.patch.set_fill(True)
ax2.patch.set_color('red')
ax2.patch.set_alpha('0.5')

trans = transforms.blended_transform_factory(ax1.transData, ax1.transAxes)
#creo un rettangolo definendo la sua larghezza in coordinate dati e la sua altezza in
coordinate axes
rect = patches.Rectangle((-0.1,0.0), width=0.2, height=1.0,
                        transform=trans, color='red',
                        alpha=0.5)

ax1.add_patch(rect)

#per note aggiuntive sulle annotazioni vedi:
#http://matplotlib.org/users/annotations_guide.html
ax1.annotate('', xy=(.775, 0.55), xycoords='axes fraction',
            xytext=(0.1, 0.95), textcoords='data',
            arrowprops=dict(edgecolor='red',arrowstyle="->",
            connectionstyle="arc3,rad=-.2",linewidth=1),
            horizontalalignment='right', verticalalignment='top',
```

```
)
```

```
pl.savefig("./img/matplotlib25.png")  
pl.show()
```

