```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
cars_data = pd.read_csv('/content/Toyota.csv',index_col=0,na_values=['??','???'])
```

```
cars_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1436 entries, 0 to 1435
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Price       1436 non-null   int64  
 1   Age         1336 non-null   float64
 2   KM          1421 non-null   float64
 3   FuelType    1336 non-null   object 
 4   HP          1436 non-null   object 
 5   MetColor    1286 non-null   float64
 6   Automatic   1436 non-null   int64  
 7   CC          1436 non-null   int64  
 8   Doors       1436 non-null   object 
 9   Weight      1436 non-null   int64  
dtypes: float64(3), int64(4), object(3)
memory usage: 123.4+ KB
```

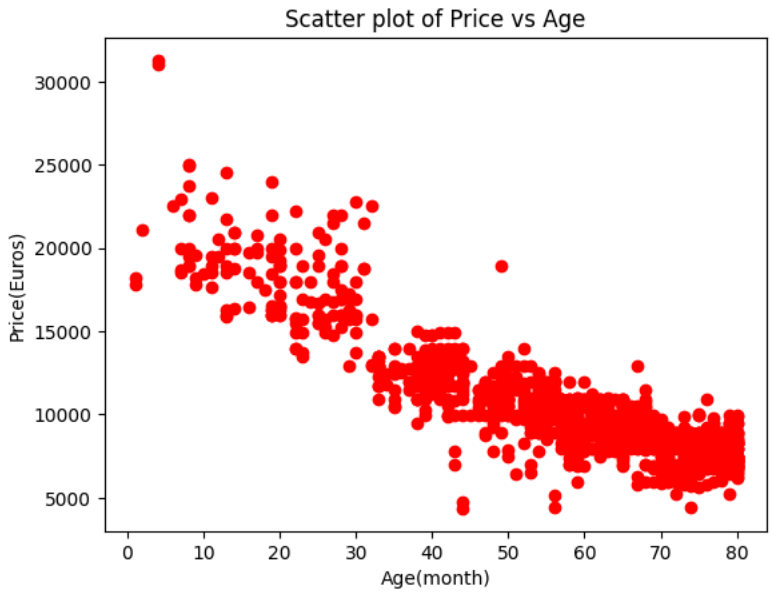
```
cars_data.dropna(axis=0,inplace=True)
cars_data
```

	Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
0	13500	23.0	46986.0	Diesel	90	1.0	0	2000	three	1165
1	13750	23.0	72937.0	Diesel	90	1.0	0	2000	3	1165
3	14950	26.0	48000.0	Diesel	90	0.0	0	2000	3	1165
4	13750	30.0	38500.0	Diesel	90	0.0	0	2000	3	1170
5	12950	32.0	61000.0	Diesel	90	0.0	0	2000	3	1170
...	...	...	...	...	...	...	...	...	...	...
1423	7950	80.0	35821.0	Petrol	86	0.0	1	1300	3	1015
1424	7750	73.0	34717.0	Petrol	86	0.0	0	1300	3	1015
1429	8950	78.0	24000.0	Petrol	86	1.0	1	1300	5	1065
1430	8450	80.0	23000.0	Petrol	86	0.0	0	1300	3	1015
1435	6950	76.0	1.0	Petrol	110	0.0	0	1600	5	1114

1099 rows × 10 columns

Scatter plot

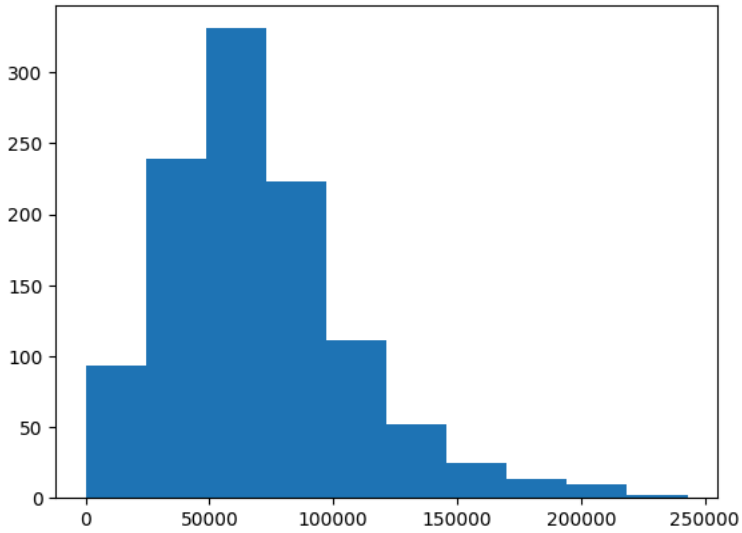
```
plt.scatter(cars_data['Age'],cars_data['Price'],c='red')
plt.title('Scatter plot of Price vs Age')
plt.xlabel('Age(month)')
plt.ylabel('Price(Euros)')
plt.show()
```



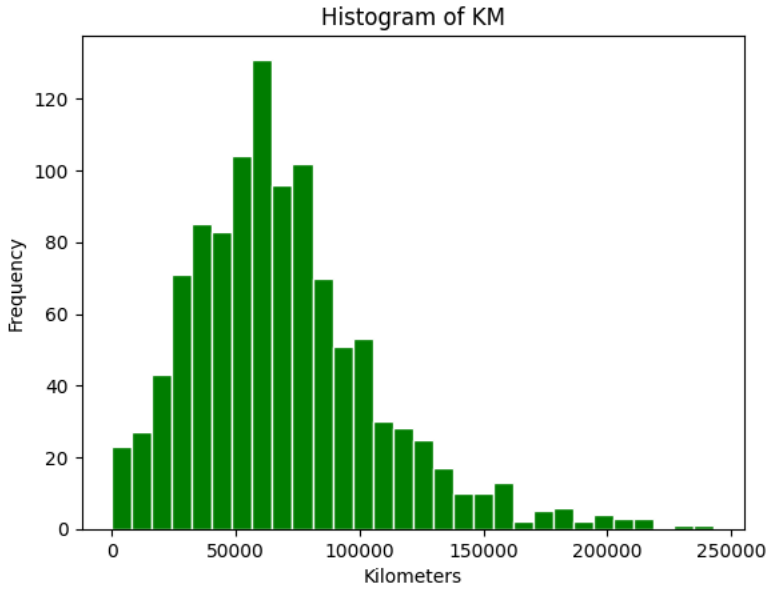
Histogram

```
plt.hist(cars_data['KM'])
```

```
(array([ 93., 239., 331., 223., 111.,  52.,  25.,  13.,  10.,   2.]),
 array([1.000000e+00, 2.430090e+04, 4.860080e+04, 7.290070e+04,
        9.720060e+04, 1.215005e+05, 1.458004e+05, 1.701003e+05,
        1.944002e+05, 2.187001e+05, 2.430000e+05]),
 <BarContainer object of 10 artists>)
```



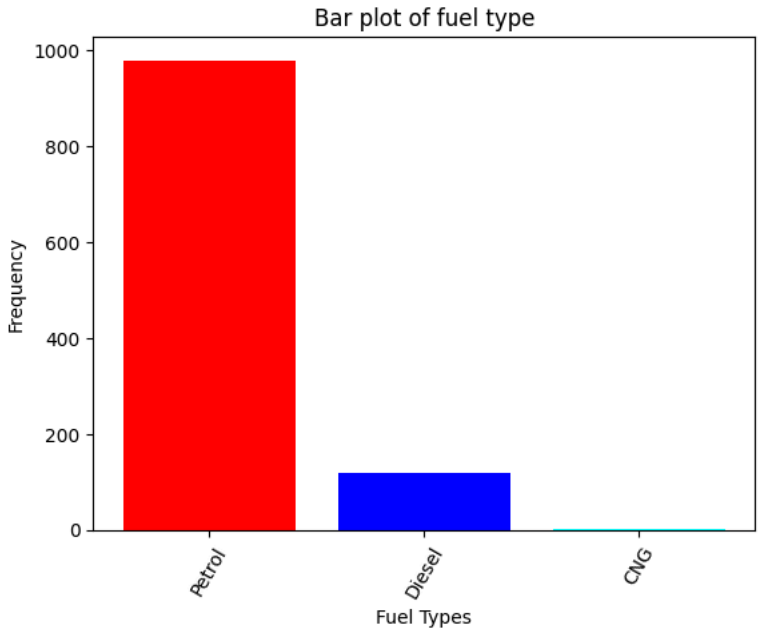

```
plt.hist(cars_data['KM'],color='green',edgecolor='white' ,bins=30)
plt.title('Histogram of KM')
plt.xlabel('Kilometers')
plt.ylabel('Frequency')
plt.show()
```



Bar plot

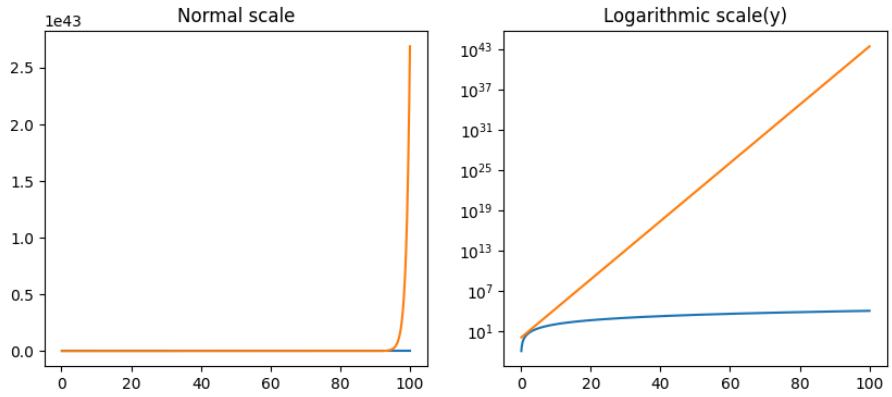
```
counts =[979,120,2]
fueltype =('Petrol','Diesel','CNG')
index=np.arange(len(fueltype))
```

```
plt.bar(index,counts,color=['red','blue','cyan'])
plt.title('Bar plot of fuel type')
plt.xlabel('Fuel Types')
plt.ylabel('Frequency')
plt.xticks(index,fueltype,rotation=60)
plt.show()
```

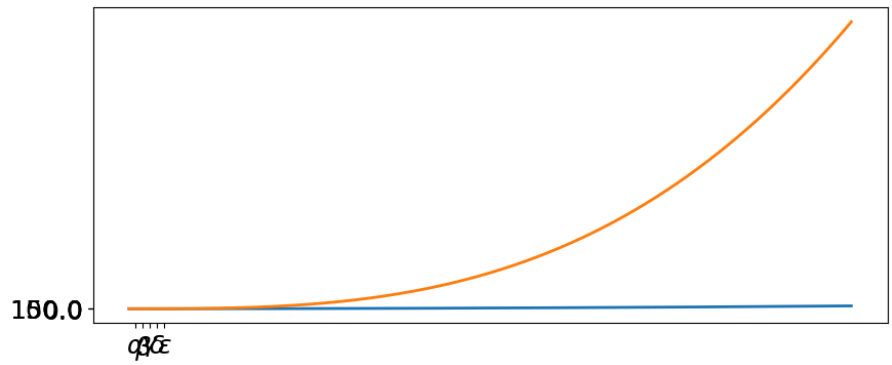


Logarithmic scale

```
fig,axes = plt.subplots(1,2,figsize=(10,4))
x=np.linspace(0.1,100,1000)

axes[0].plot(x,x**2,x,np.exp(x))
axes[0].set_title('Normal scale')

axes[1].plot(x,x**2,x,np.exp(x))
axes[1].set_yscale('log')
axes[1].set_title('Logarithmic scale(y)');
```

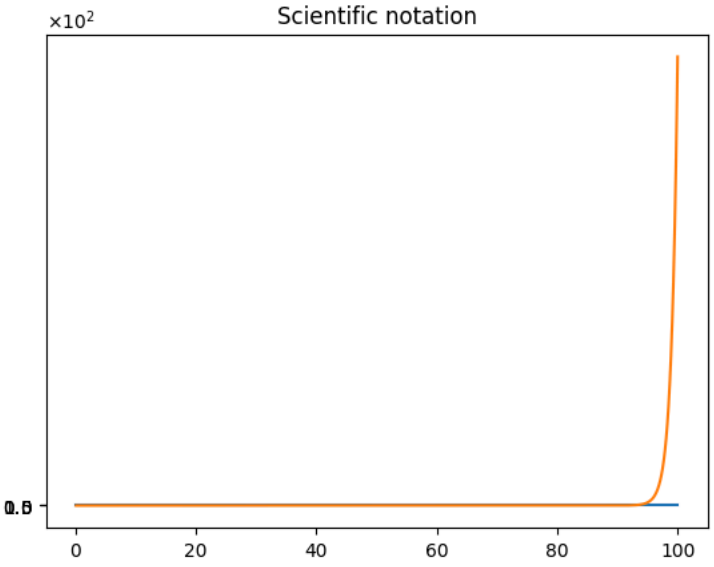



```
fig, ax=plt.subplots(figsize=(10,4))

ax.plot(x, x**2, x, x**3, lw=2)

ax.set_xticks([1,2,3,4,5])
ax.set_xticklabels([r'$\alpha$',r'$\beta$',r'$\gamma$',r'$\delta$','$\epsilon$'], fontsize=18)

yticks=[0, 50, 100, 150]
ax.set_yticks(yticks)
ax.set_yticklabels(["$%.1f$" % y for y in yticks], fontsize=18);
```



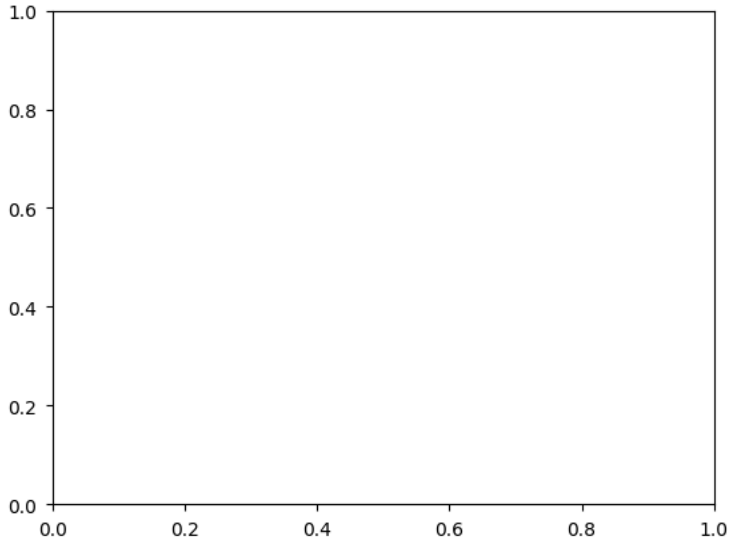
Scientific notation

```
fig, ax = plt.subplots(1,1)

ax.plot(x,x**2, x, np.exp(x))
ax.set_title("Scientific notation")

ax.set_yticks([0,50,100,150])

from matplotlib import ticker
formatter = ticker.ScalarFormatter(useMathText=True)
formatter.set_scientific(True)
formatter.set_powerlimits((-1,1))
ax.yaxis.set_major_formatter(formatter)
```

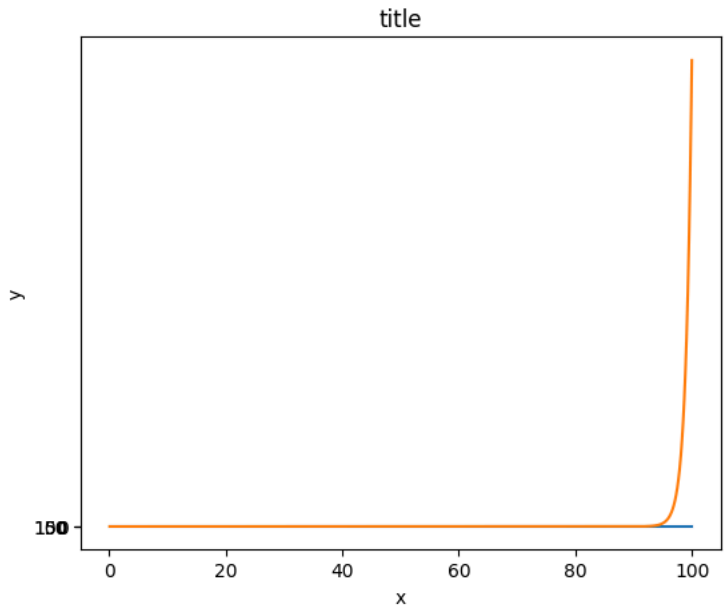


Axis number and axis number places

```
import matplotlib
matplotlib.rcParams['xtick.major.pad'] = 5
matplotlib.rcParams['ytick.major.pad'] = 5

fig.ax = plt.subplots(1,1)

ax.plot(x, x**2, x, np.exp(x))
ax.set_yticks([0, 50, 100, 150])
ax.set_title("Label and axis spacing")
# pading between axis label and axis numbers
ax.xaxis.labelpad = 5
ax.yaxis.labelpad = 5
ax.set_xlabel("x")
ax.set_ylabel("y");
```

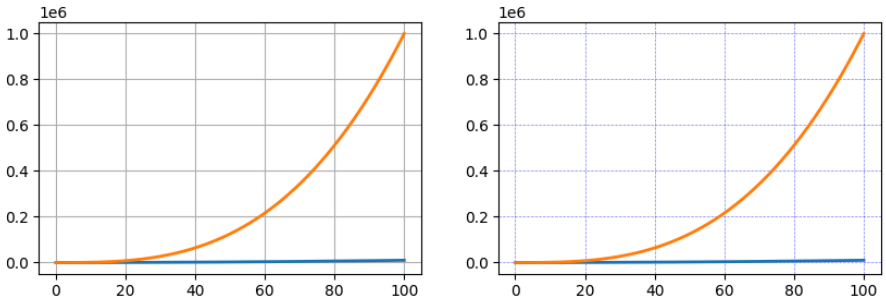



```
#restore defaults
matplotlib.rcParams['xtick.major.pad'] = 3
matplotlib.rcParams['ytick.major.pad'] = 3


fig,ax=plt.subplots(1, 1)

ax.plot(x, x**2,x, np.exp(x))
ax.set_yticks([0, 50, 100, 150])

ax.set_title("title")
ax.set_xlabel("x")
ax.set_ylabel("y")

fig.subplots_adjust(left=0.15,right=.9,bottom=0.1, top=0.9);
```



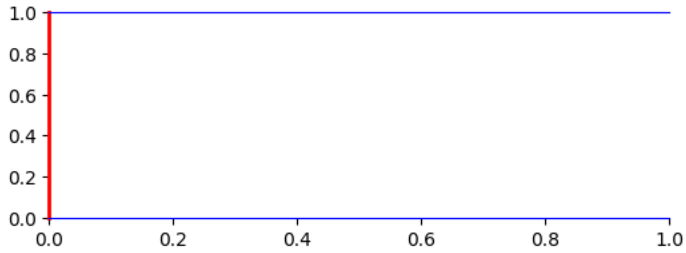
Axis grid

```
fig, axes = plt.subplots(1,2, figsize=(10,3))
# default grid appearance
axes[0].plot(x, x**2, x, x**3, lw=2)
axes[0].grid(True)
# custom grid apperance
axes[1].plot(x, x**2, x, x**3, lw=2)
axes[1].grid(color='b', alpha=0.5, linestyle='dashed', linewidth=0.5)
```

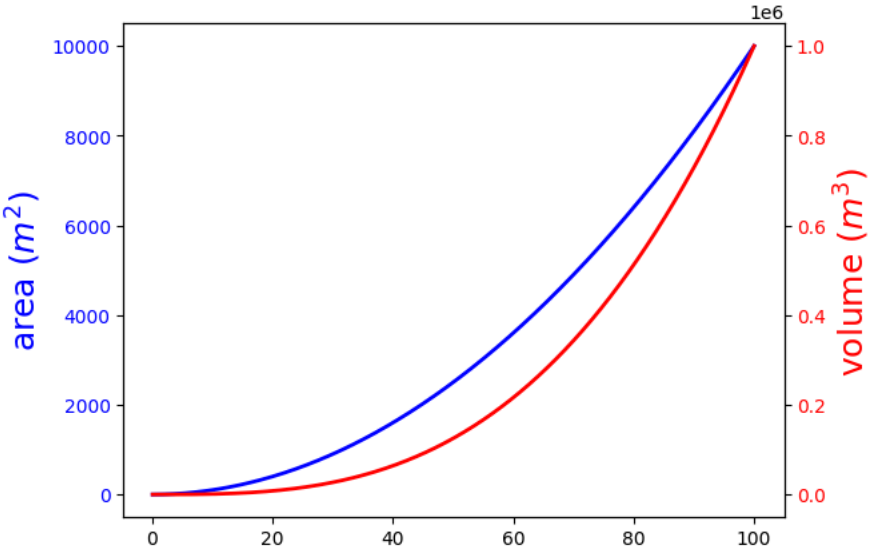


Axis spines

```
fig, ax =plt.subplots(figsize=(6,2))
ax.spines['bottom'].set_color('blue')
ax.spines['top'].set_color('blue')
ax.spines['left'].set_color('red')
ax.spines['left'].set_linewidth(2)
# turn off axis spine to the right
ax.spines['right'].set_color('none')
ax.yaxis.tick_left()
```

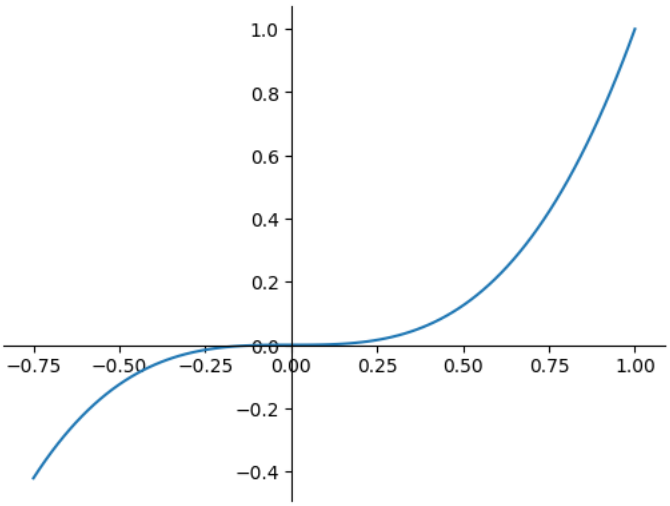



```
fig, ax1=plt.subplots()

ax1.plot(x, x**2, lw=2, color='blue')
ax1.set_ylabel(r"area $(m^2)$", fontsize=18,color='blue')
for label in ax1.get_yticklabels():
  label.set_color("blue")

ax2 = ax1.twinx()
ax2.plot(x, x**3, lw=2, color='red')
ax2.set_ylabel(r"volume $(m^3)$", fontsize=18,color='red')
for label in ax2.get_yticklabels():
  label.set_color('red')
```



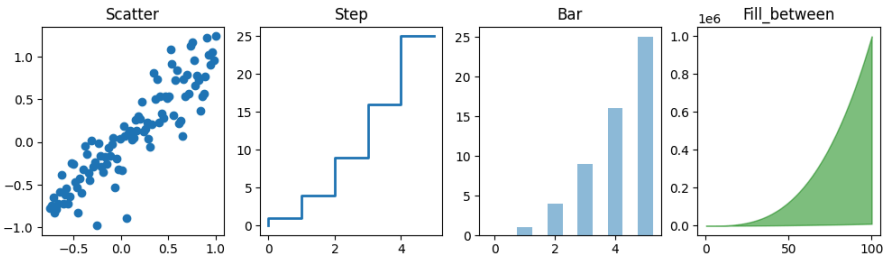
Axis where x and y is zero

```
fig, ax = plt.subplots()

ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')

ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0)) # set position of x spines

ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0)) # set position of y spines

xx = np.linspace(-0.75, 1., 100)
ax.plot(xx, xx**3);
```



2d plots

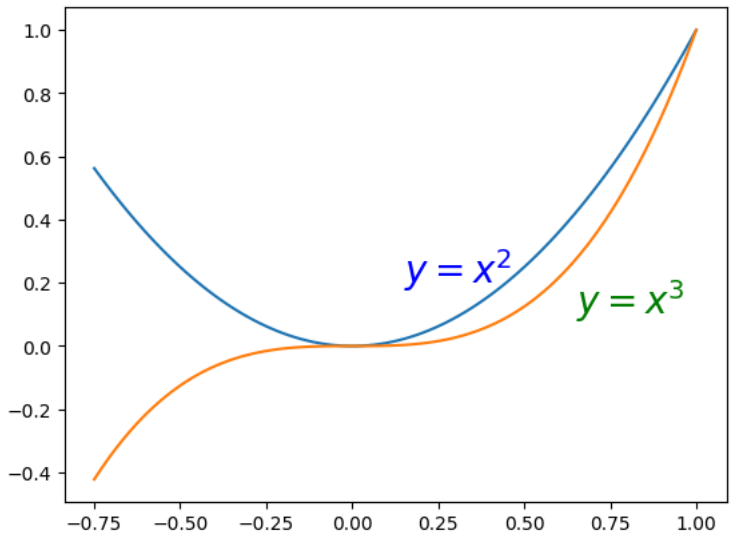
```
n= np.array([0,1,2,3,4,5])

fig, axes = plt.subplots(1, 4, figsize=(12,3))

axes[0].scatter(xx, xx+0.25*np.random.randn(len(xx)))
axes[0].set_title("Scatter")

axes[1].step(n, n**2, lw=2)
axes[1].set_title("Step")


axes[2].bar(n, n**2,align='center', width=0.5, alpha=0.5)
axes[2].set_title("Bar")


axes[3].fill_between(x, x**2, x**3,  color='green', alpha=0.5);
axes[3].set_title("Fill_between");
```



Text annonation

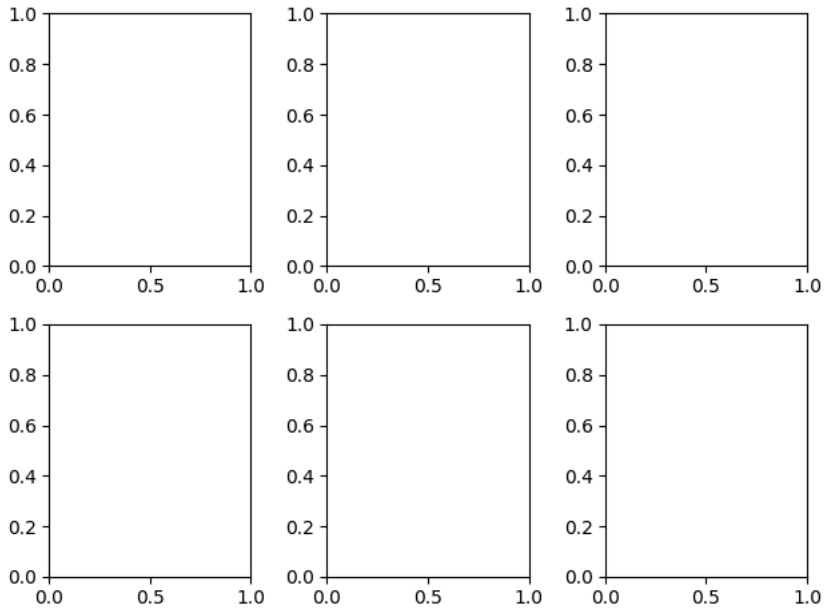
```
fig, ax = plt.subplots()

ax.plot(xx, xx**2, xx ,xx**3)

ax.text(0.15,0.2, r"$y=x^2$", fontsize=20, color="blue")
ax.text(0.65,0.1, r"$y=x^3$", fontsize=20, color="green")
```

```
Text(0.65, 0.1, '$y=x^3$')
```

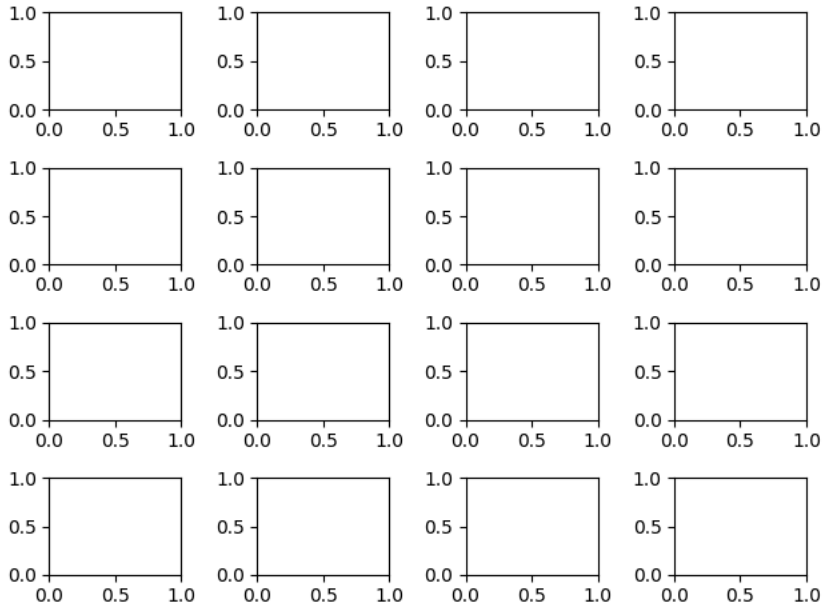


Figures with multiple subplots and inserts

```
fig, ax=plt.subplots(2,3)
fig.tight_layout()
```

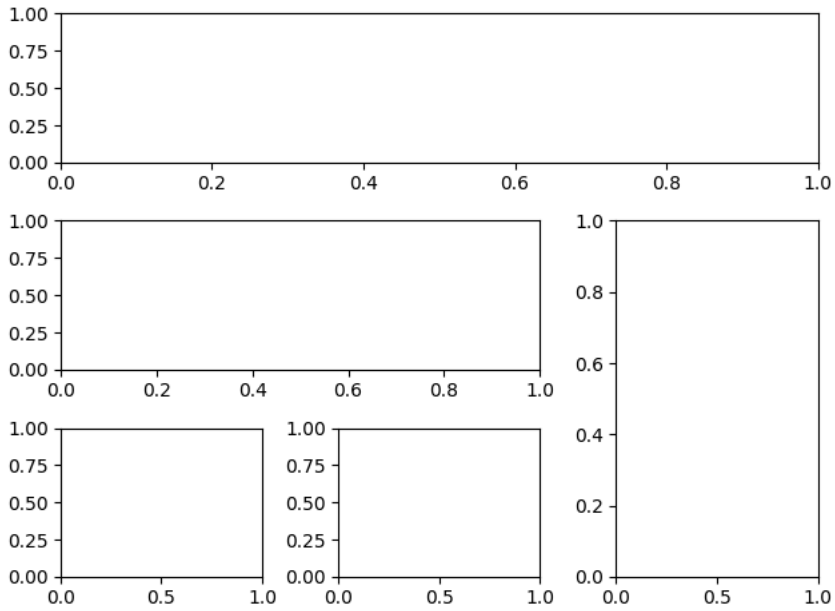


```
fig, ax=plt.subplots(4,4)
fig.tight_layout()
```



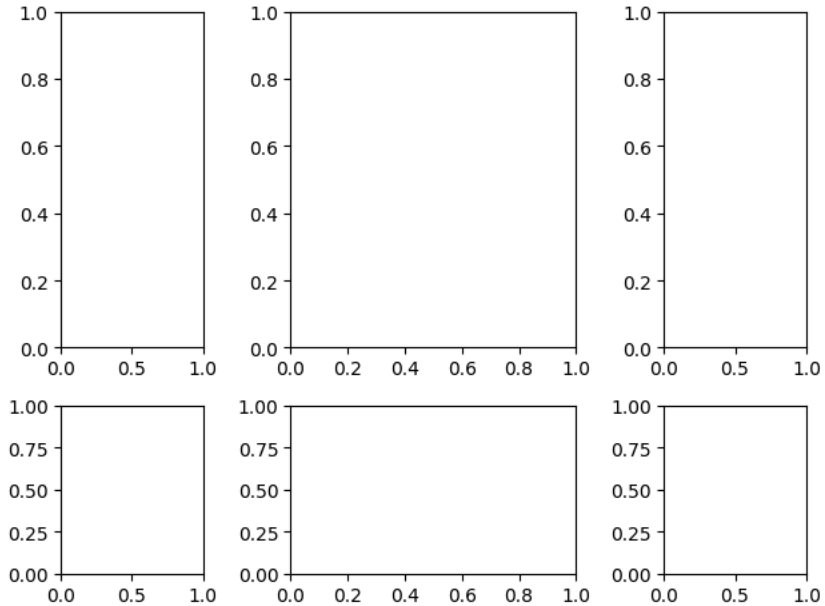
subplot 2 grid

```
fig = plt.figure()
ax1 = plt.subplot2grid((3,3),(0,0), colspan=3)
ax2 = plt.subplot2grid((3,3),(1,0), colspan=2)
ax3 = plt.subplot2grid((3,3),(1,2), rowspan=2)
ax4 = plt.subplot2grid((3,3),(2,0))
ax5 = plt.subplot2grid((3,3),(2,1))
fig.tight_layout()
```



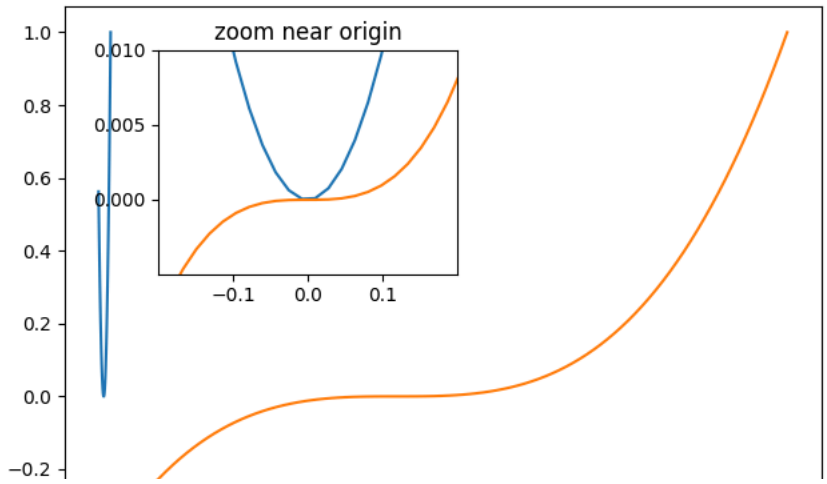
grid spec

```
import matplotlib.gridspec as gridspec

fig = plt.figure()

gs = gridspec.GridSpec(2, 3, height_ratios=[2,1], width_ratios=[1,2,1])
for g in gs:
  ax=fig.add_subplot(g)

fig.tight_layout()
```



add axes

```
fig, ax = plt.subplots()

ax.plot(xx, xx**2, xx**3)
fig.tight_layout()

#inset
inset_ax = fig.add_axes([0.2, 0.55, 0.35, 0.35]) #X,Y, width, height

inset_ax.plot(xx, xx**2, xx, xx**3)
inset_ax.set_title('zoom near origin')

# set axis range
inset_ax.set_xlim(-.2, .2)
inset_ax.set_ylim(-.005, .01)

#set axis tick locations
inset_ax.set_yticks([0, 0.005, 0.01])
inset_ax.set_xticks([-0.1,0,.1]);
```



color map and countour figures

```
alpha = 0.7
phi_ext = 2*np.pi*0.5
def flux_qubit_potential(phi_m, phi_p):
```