```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
import numpy as np
x = np.linspace(0,5,11)
y = x ** 2
```

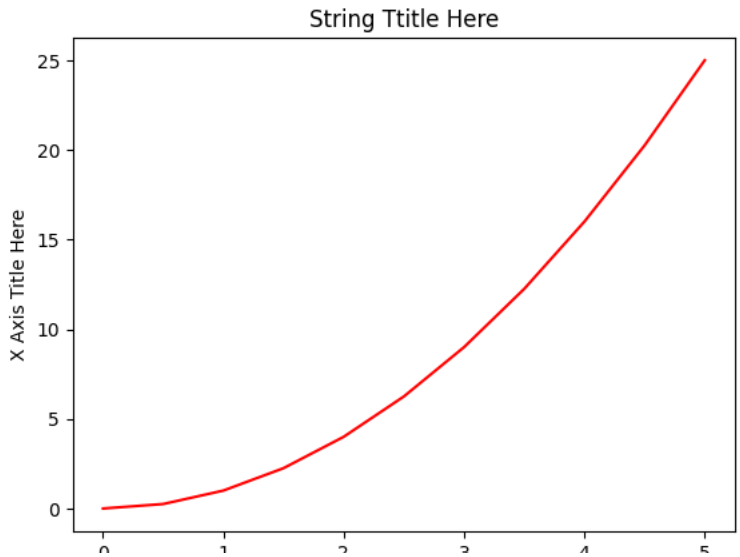
```
x
```

```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

```
y
```

```
array([ 0.  ,  0.25,  1.  ,  2.25,  4.  ,  6.25,  9.  , 12.25, 16.  ,
       20.25, 25.  ])
```

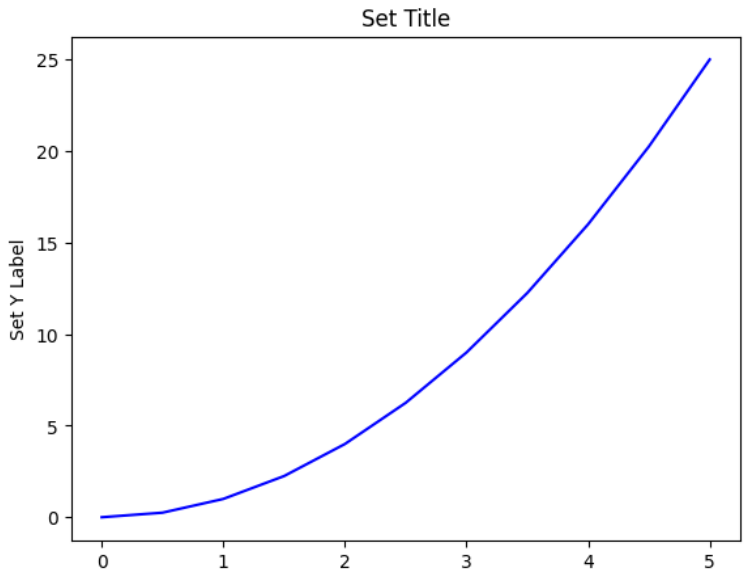
```
plt.plot(x, y, 'r') #'r' is the color red
plt.xlabel('X Axis Title Here')
plt.ylabel('X Axis Title Here')
plt.title('String Ttitle Here')
plt.show()
```




```
#plt.subplot(nrows, ncols, plot_number)
plt.subplot(1,2,1)
plt.plot(x ,y, 'r--') #More on color options later
plt.subplot(1,2,2)
plt.plot(y, x, 'g*-');
```

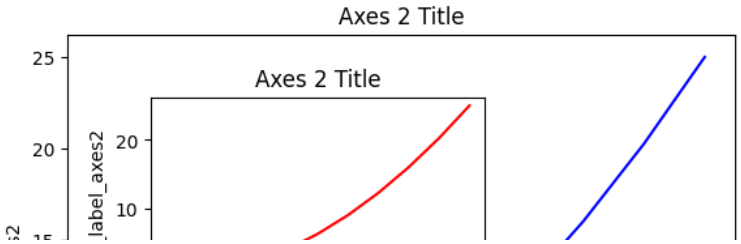

```
#Creat Figure (empty canvas)
fig = plt.figure()

#Add set of axes to figure
axes = fig.add_axes([0.1,0.1,0.8,0.8]) #Left,bottom,width,height

#Flot on that set of axes
axes.plot(x, y, 'b')
axes.set_xlabel('Set X Label') #Notice the use of set_ to begin methods
axes.set_ylabel('Set Y Label')
axes.set_title('Set Title')
```

```
Text(0.5, 1.0, 'Set Title')
```



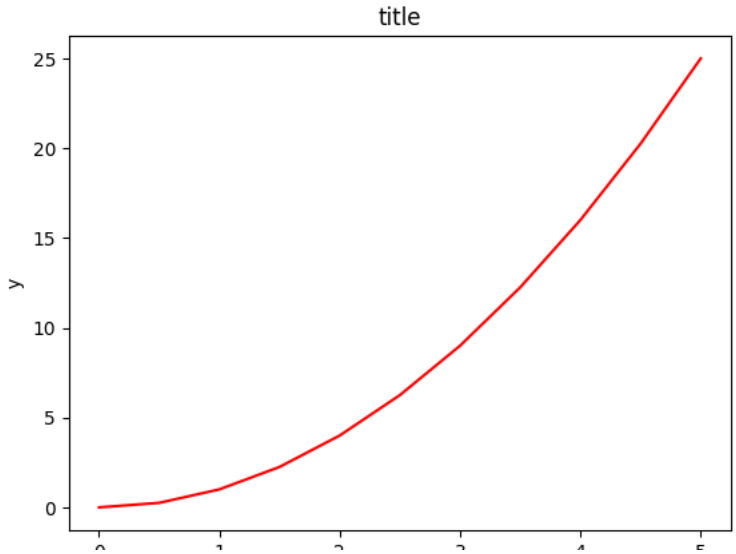

```
#Creat blank canvas
fig=plt.figure()

#Add set of axes to figure
axes1=fig.add_axes([0.1,0.1,0.8,0.8]) #Main axes
axes2=fig.add_axes([0.2,0.5,0.4,0.3]) #Inset axes

#Larger Figure Axes 1
axes1.plot(x, y, 'b')
axes1.set_xlabel('X_label_axes2')
axes1.set_ylabel('Y_label_axes2')
axes1.set_title('Axes 2 Title')

#Larger Figure Axes 2
axes2.plot(x, y, 'r')
axes2.set_xlabel('X_label_axes2')
axes2.set_ylabel('Y_label_axes2')
axes2.set_title('Axes 2 Title')
```

```
Text(0.5, 1.0, 'Axes 2 Title')
```

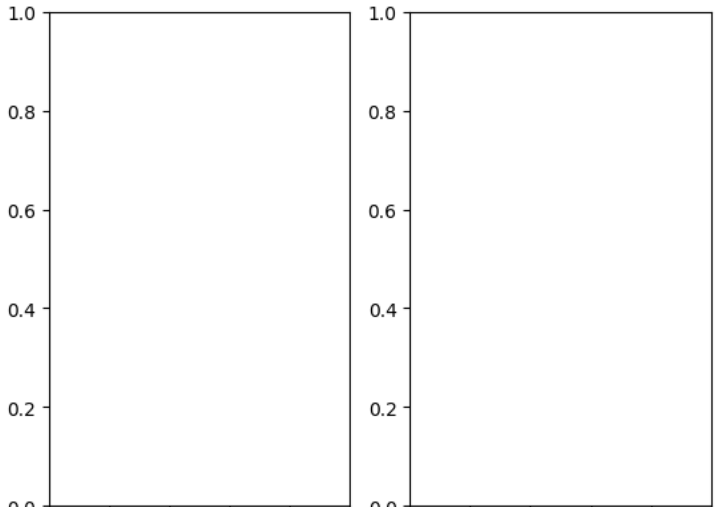



```
#Use similar to plt.figure() except use tuple unpacking to grab fig and axes
fig, axes = plt.subplots()

#Now use the axes object ot add stuff to plot
axes.plot(x ,y, 'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title')
```

```
Text(0.5, 1.0, 'title')
```



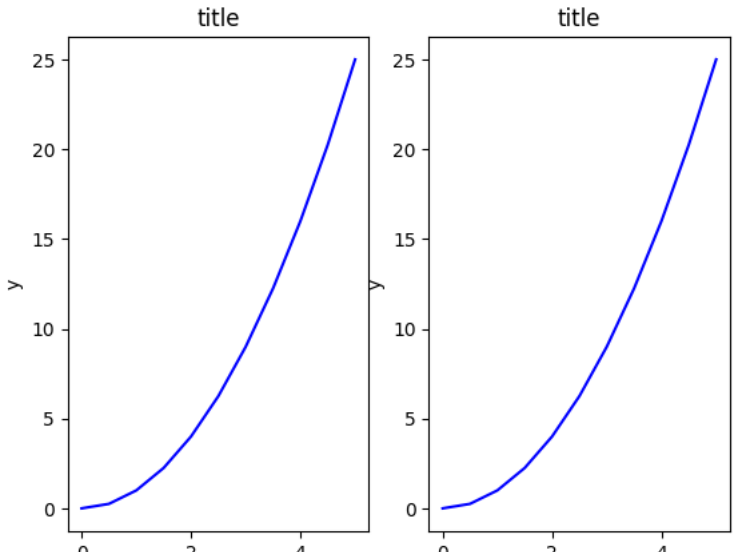

```
#Empty canvas of 1 by 2 subplots
fig, axes = plt.subplots(nrows=1, ncols=2)
```



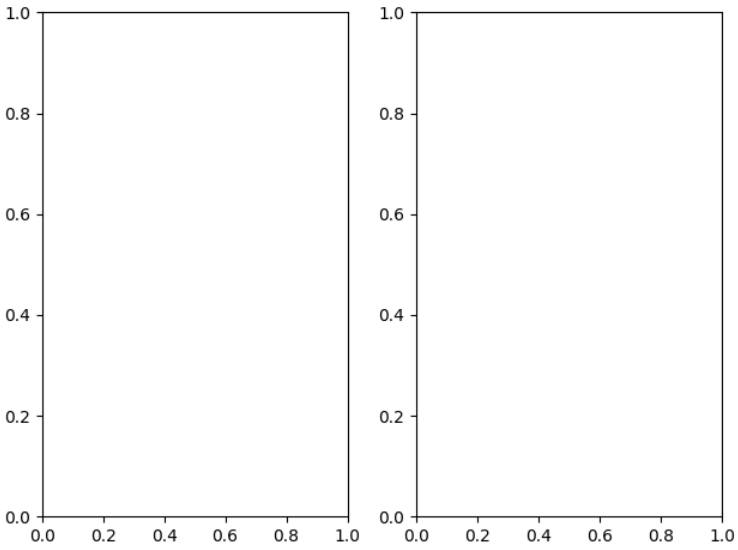
```
#Axes is an array of axes to plot on
axes
```

```
array([<Axes: >, <Axes: >], dtype=object)
```

```
for ax in axes:
  ax.plot(x, y, 'b')
  ax.set_xlabel('x')
  ax.set_ylabel('y')
  ax.set_title('title')

#Display the figure object
fig
```



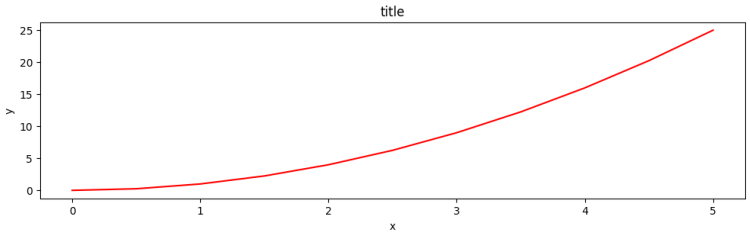

```
fig, axe = plt.subplots(nrows=1, ncols=2)

for ax in axes:
  ax.plot(x, y, 'b')
  ax.set_xlabel('x')
  ax.set_ylabel('y')
  ax.set_title('title')

fig
plt.tight_layout()
```




```
fig = plt.figure(figsize=(8, 4), dpi = 100)
```

```
<Figure size 800x400 with 0 Axes>
```

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

axes.plot(x, y, 'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```




```
fig.savefig("filename.png")
```

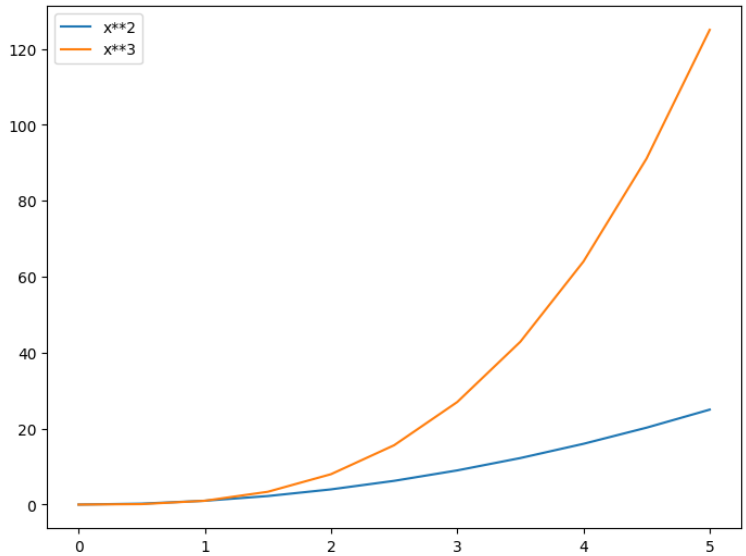
```
fig.savefig("filename.png", dpi = 200)
```

```
ax.set_title("title");
```

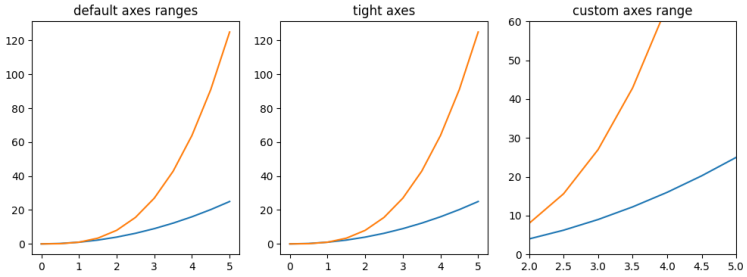
```
ax.set_xlabel("x")
ax.set_ylabel("y");
```

```
fig = plt.figure()

ax = fig.add_axes([0, 0, 1, 1])

ax.plot(x, x**2, label = "x**2")
ax.plot(x, x**3, label = "x**3")
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7f1d0dbfe5e0>
```



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

axes[0].plot(x, x**2, x, x**3)
axes[0].set_title("default axes ranges")

axes[1].plot(x, x**2, x, x**3)
axes[1].axis('tight')
axes[1].set_title("tight axes")

axes[2].plot(x, x**2, x, x**3)
axes[2].set_ylim([0, 60])
axes[2].set_xlim([2, 5])
axes[2].set_title("custom axes range");
```



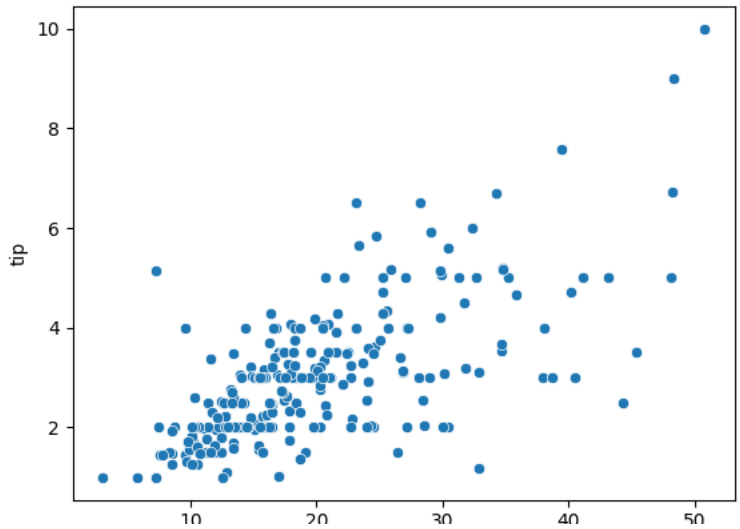

```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline

sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
tips = sns.load_dataset("tips")
tips.head()
```

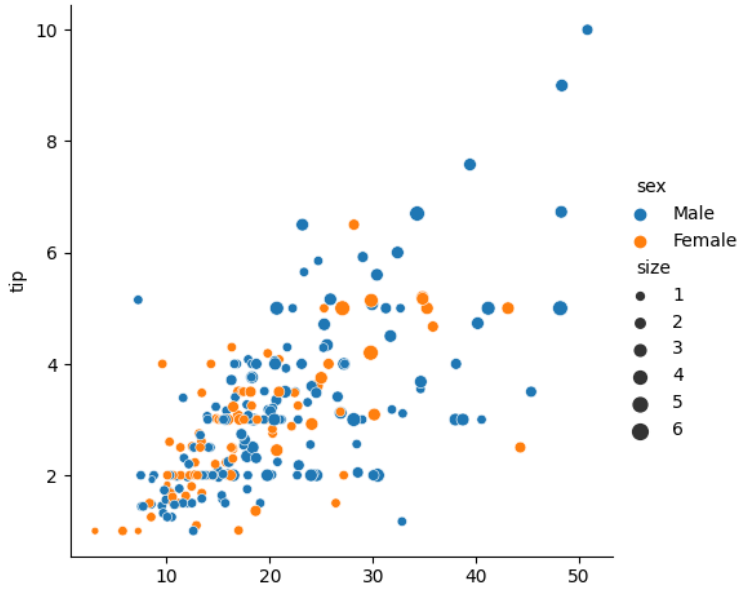
| total_bill | tip | sex | smoker | day | time | size |
|---|---|---|---|---|---|---|

```
ax = sns.scatterplot(x = "total_bill", y = "tip", data = tips)
```




```
sns.relplot(x = "total_bill", y = "tip", data = tips, kind = "scatter", hue = "sex", size = "size",)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1cfb360550>
```

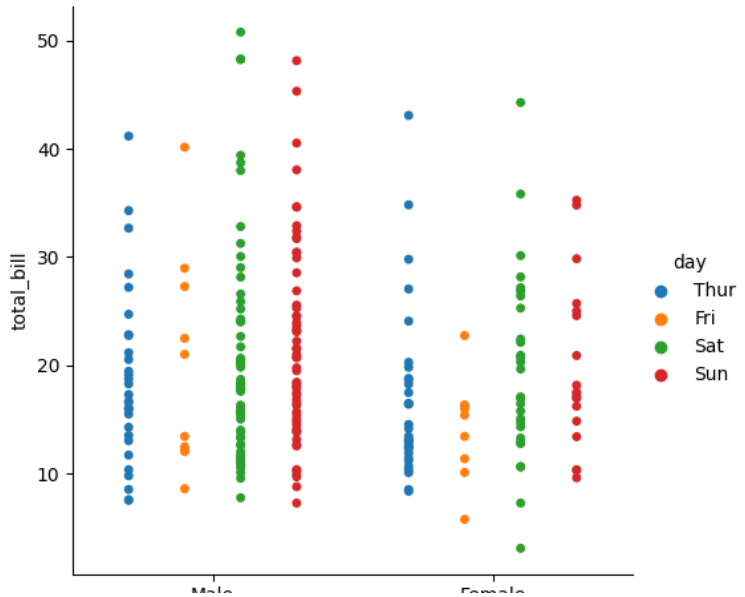



```
sns.catplot(x = "sex", y = "total_bill", hue = "day", data = tips, kind = "strip")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1cfb1f53a0>
```

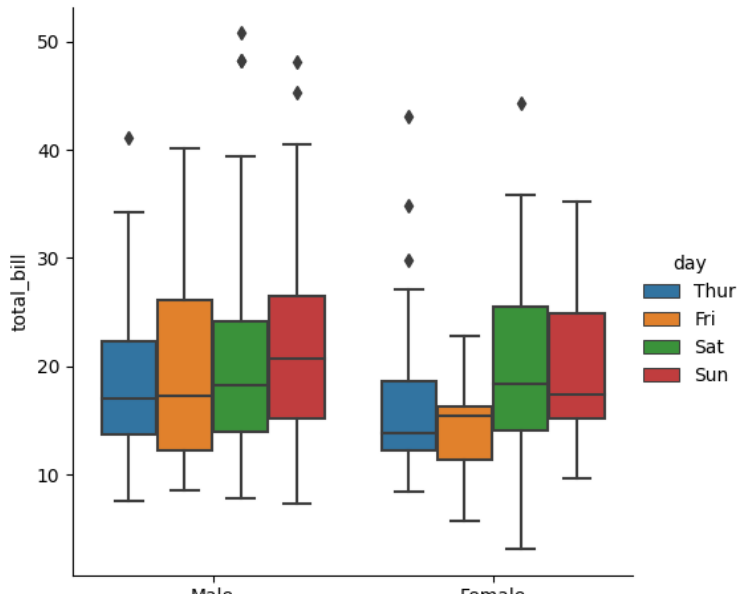

```
sns.catplot(x = "sex", y = "total_bill", hue = "day", data = tips, kind = "strip", jitter = False, dodge = True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1cfb43d8e0>
```




```
sns.catplot(x ="sex", y = "total_bill", hue = "day", data = tips, kind = "box")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f1cfb3b29a0>
```




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

```
data = pd.read_csv('job-market.csv')
job_counts = data.groupby('Classification').count()['Id']
job_counts = job_counts.sort_values(ascending=False)
colors = plt.cm.coolwarm_r(np.linspace(0, 1, len(job_counts)))

fig, ax = plt.subplots()
ax.barh(job_counts.index, job_counts.values, color=colors)
ax.invert_yaxis()
ax.set_xlabel('Number of Jobs')
ax.set_title('Job Distribution by Classification')

plt.show()
```

```
---------------------------------------------------------------------------
--
FileNotFoundError                         Traceback (most recent call
last)
<ipython-input-29-74d492f07588> in <cell line: 5>()
      3 import numpy as np
      4
----> 5 data = pd.read_csv('job-market.csv')
      6 job_counts = data.groupby('Classification').count()['Id']
      7 job_counts = job_counts.sort_values(ascending=False)

 6 frames
/usr/local/lib/python3.9/dist-packages/pandas/io/common.py in
get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,
errors, storage_options)
    854         if ioargs.encoding and "b" not in ioargs.mode:
    855             # Encoding
--> 856             handle = open(
    857                 handle,
    858                 ioargs.mode,
```

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data from a CSV file
data = pd.read_csv('job-market.csv')

# Create a new column that represents the salary range
data['Salary Range'] = data['LowestSalary'].astype(str) + '-' + data['HighestSalary'].astype(str)

# Group the data by salary range and count the number of job posts in each range
salary_counts = data.groupby('Salary Range').count()['Id']

# Create a pie chart of job posts by salary range
fig, ax = plt.subplots()
ax.pie(salary_counts.values, labels=salary_counts.index, autopct='%1.1f%%')

# Add a title to the chart
ax.set_title('Job Posts by Salary Range')

# Show the chart
plt.show()
```

0 giây hoàn thành lúc 10:31