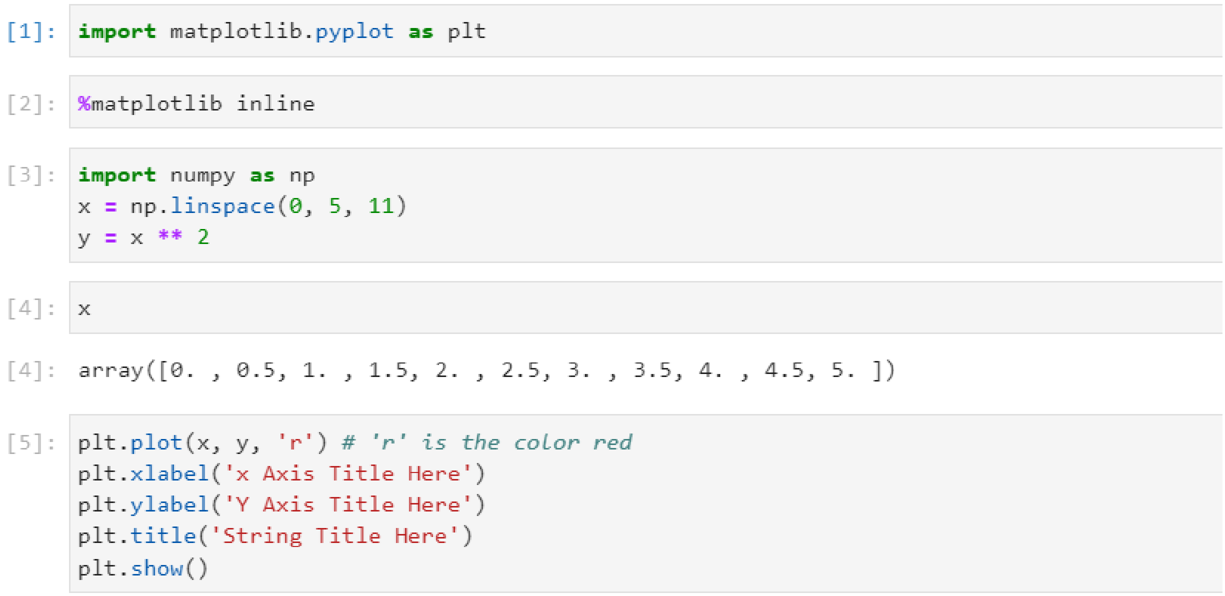
Name: Nguyễn Thanh Lộc

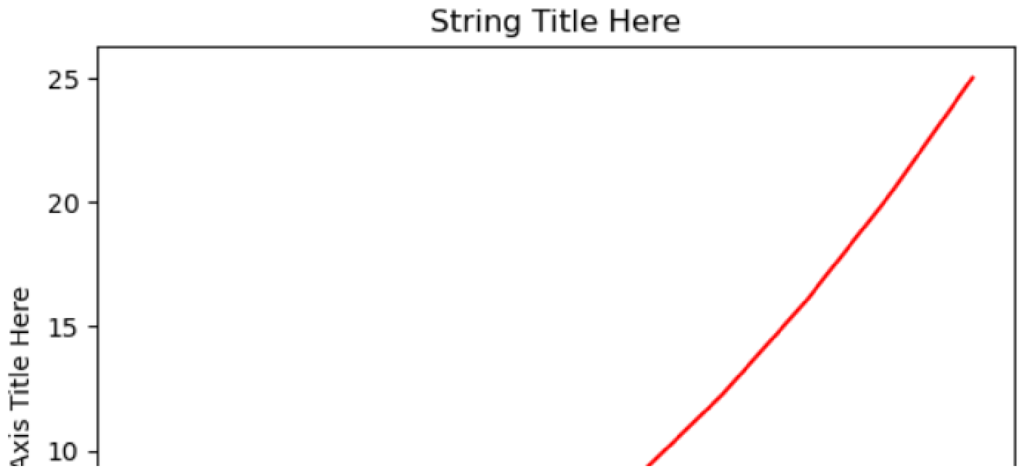
ID: 19521763

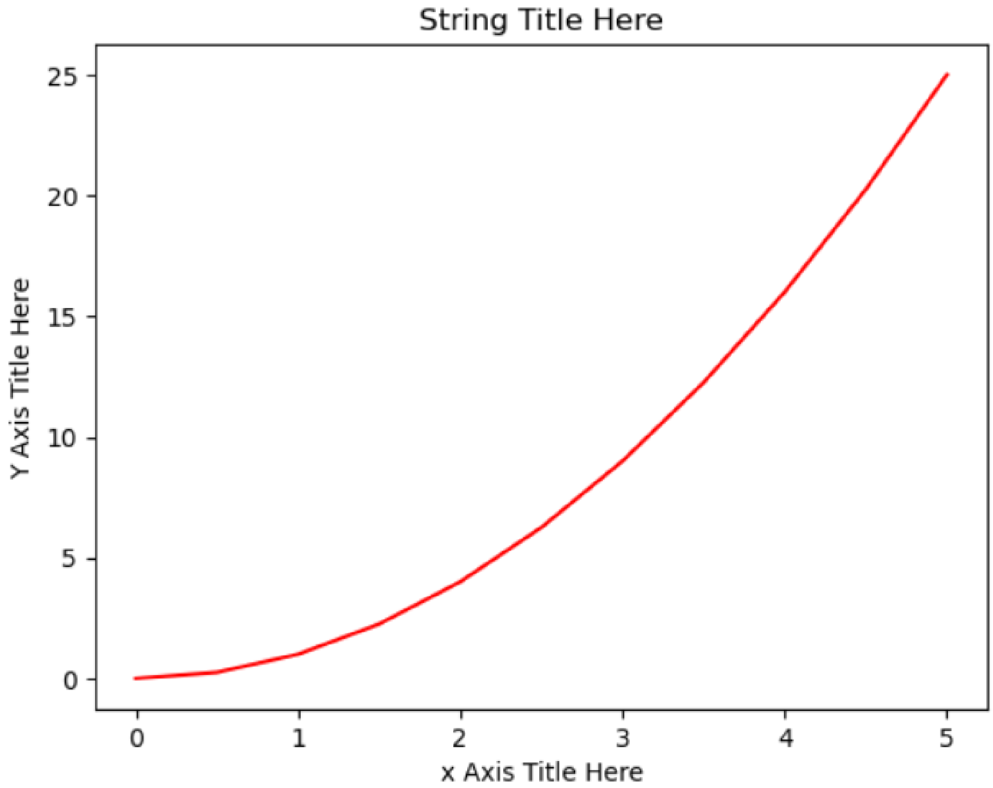
Lab3

Link github: https://github.com/talo33/Lab3



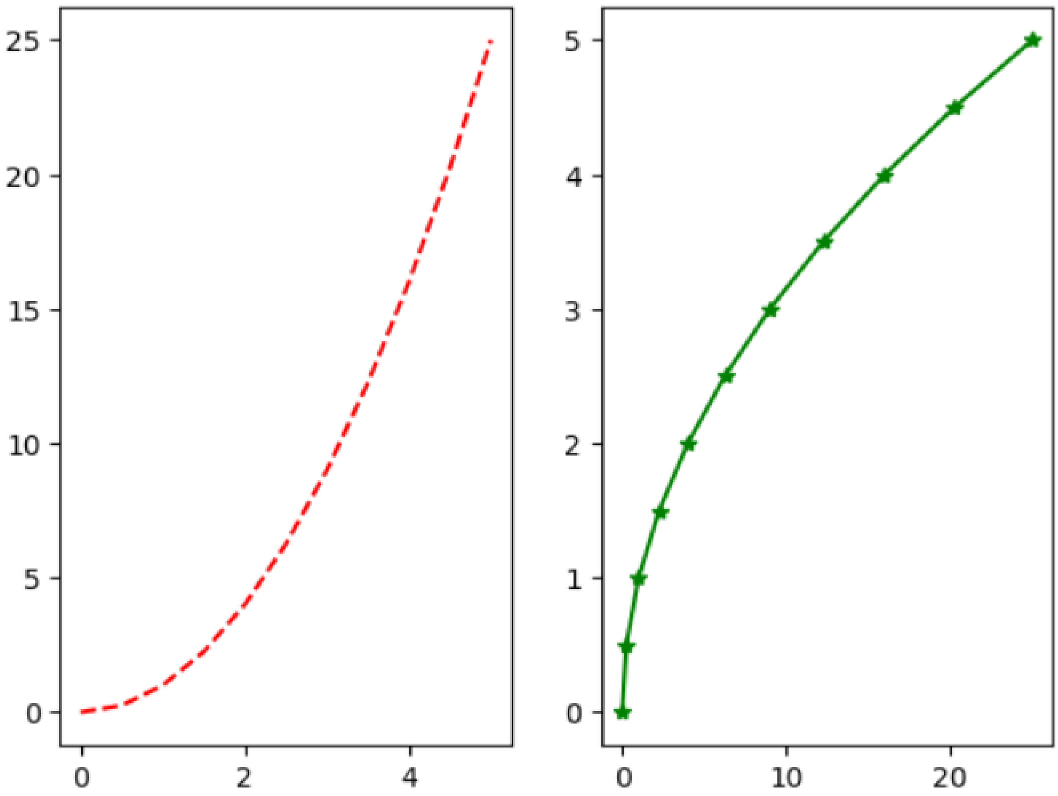
```
[1]: import matplotlib.pyplot as plt

[2]: %matplotlib inline

[3]: import numpy as np
     x = np.linspace(0, 5, 11)
     y = x ** 2

[4]: x

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

[5]: plt.plot(x, y, 'r') # 'r' is the color red
     plt.xlabel('x Axis Title Here')
     plt.ylabel('Y Axis Title Here')
     plt.title('String Title Here')
     plt.show()
```





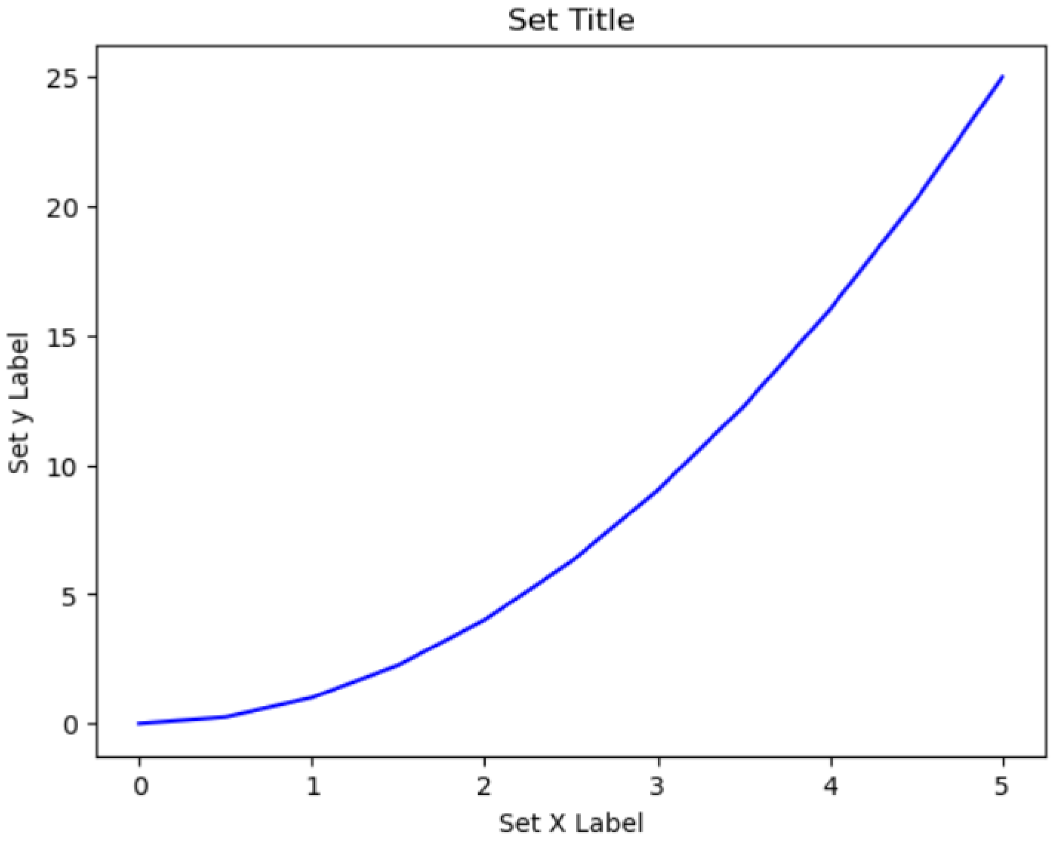
[8]:

```
# 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*-');
```

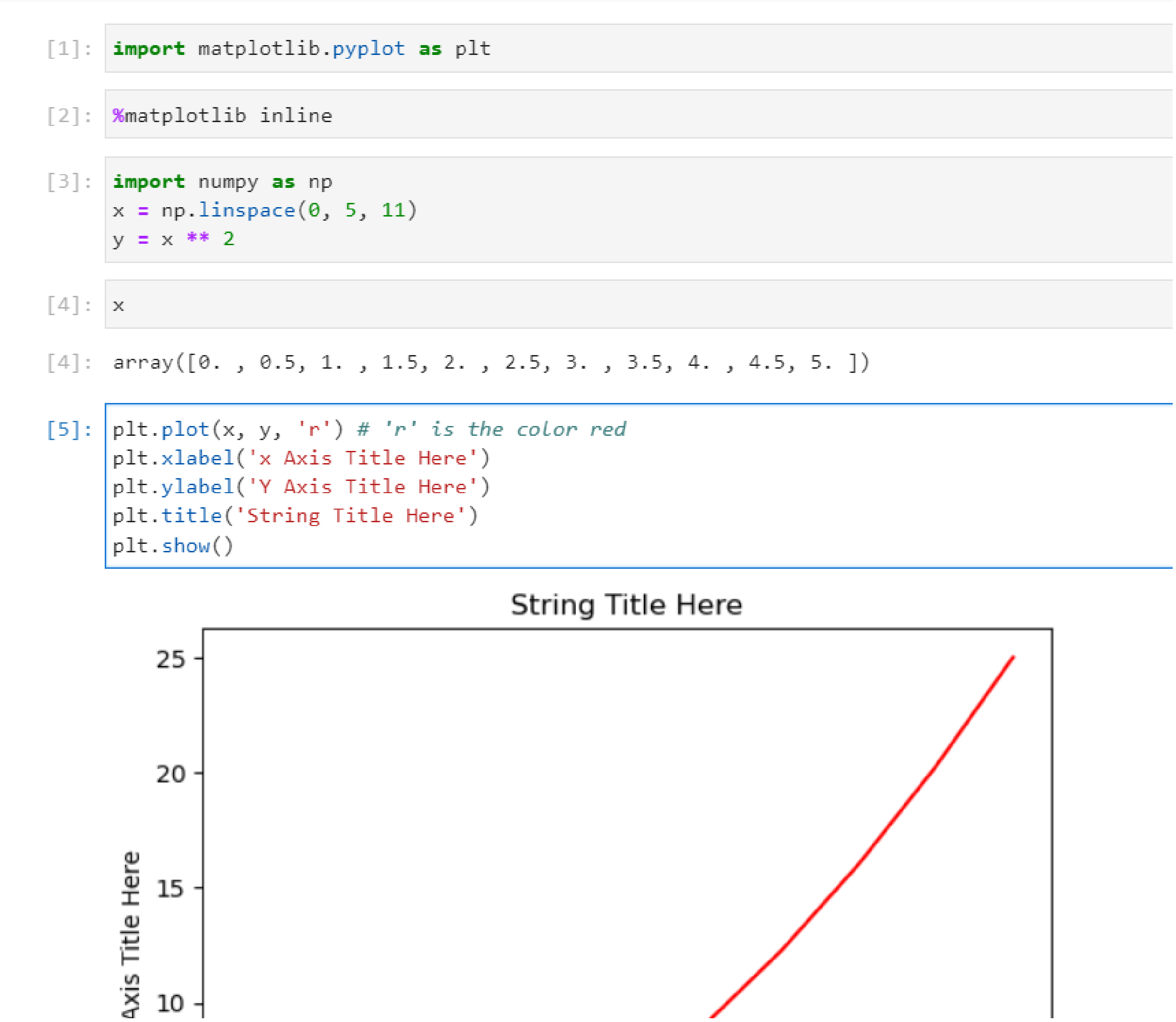


```
[11]: # Create Figure (empty canvas)
      fig = plt.figure()
      # Add set of axes to figure
      axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height
      axes.plot(x, y, 'b')
      axes.set_xlabel('Set X Label') # Notice the use Of set to begin methods
      axes.set_ylabel('Set y Label')
      axes.set_title('Set Title')
```

```
[11]: Text(0.5, 1.0, 'Set Title')
```



```
[24]: # Creates blank canvas
      fig = plt.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
```



```
[1]: import matplotlib.pyplot as plt
```

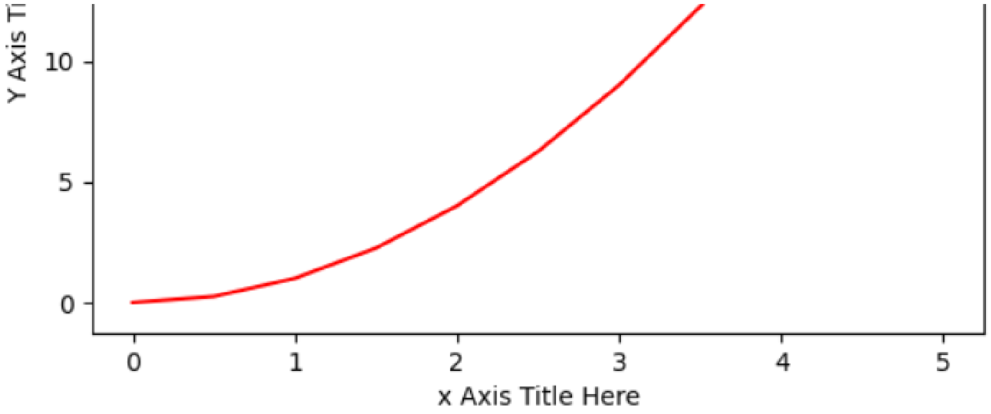
```
[2]: %matplotlib inline
```

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

```
[4]: x
```

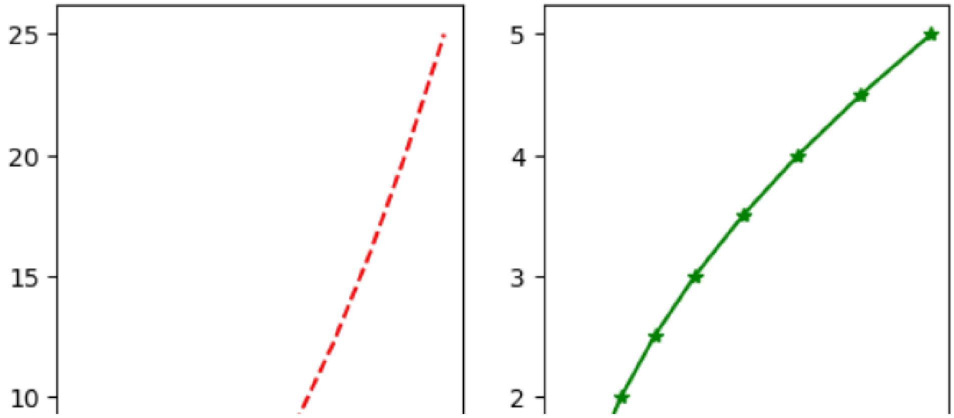
```
[4]: array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

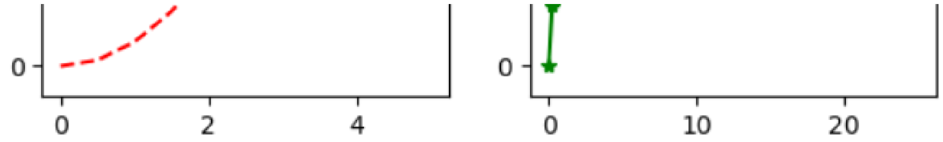
```
[5]: plt.plot(x, y, 'r') # 'r' is the color red
     plt.xlabel('x Axis Title Here')
     plt.ylabel('Y Axis Title Here')
     plt.title('String Title Here')
     plt.show()
```



[8]:

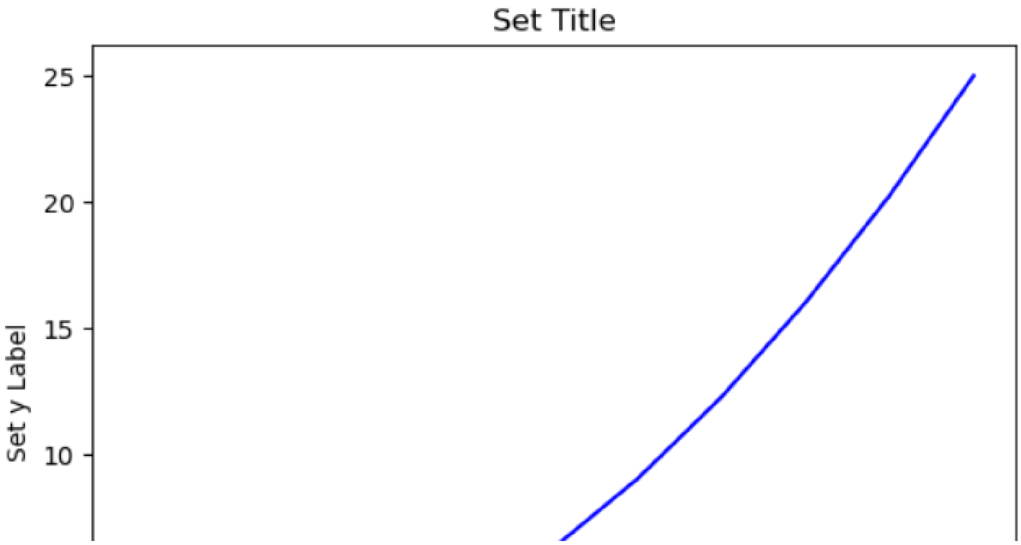
```
# plt.subplot(nrows, ncols, plot number )
plt.subplot(1,2,1)
plt.plot(x, y, 'r--') # More on color options later
plt.subplot(1,2,2)
plt.plot(y, x, 'g*-');
```

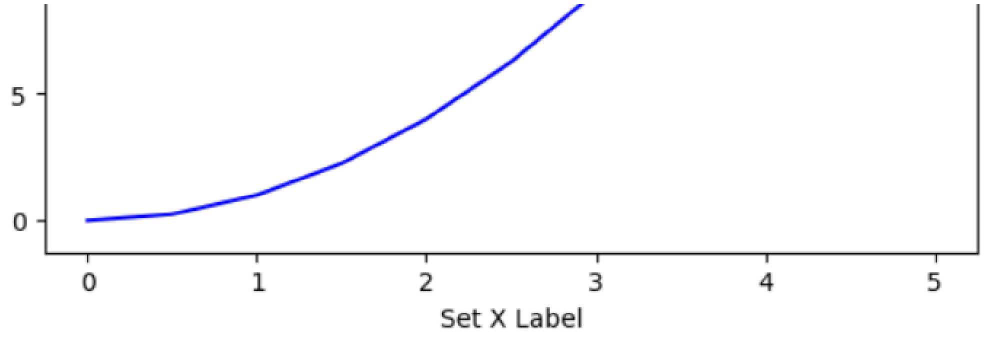


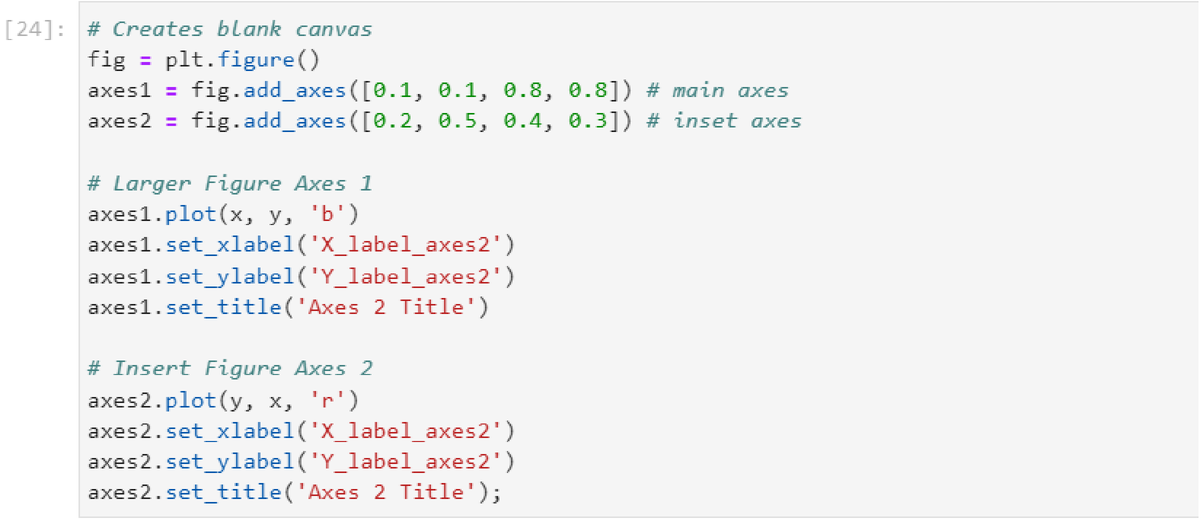


```
[11]: # Create Figure (empty canvas)
      fig = plt.figure()
      # Add set of axes to figure
      axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height
      axes.plot(x, y, 'b')
      axes.set_xlabel('Set X Label') # Notice the use Of set to begin methods
      axes.set_ylabel('Set y Label')
      axes.set_title('Set Title')
```

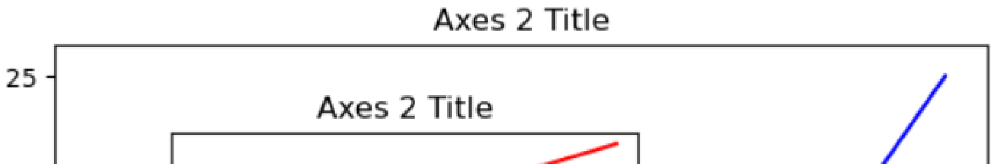
```
[11]: Text(0.5, 1.0, 'Set Title')
```

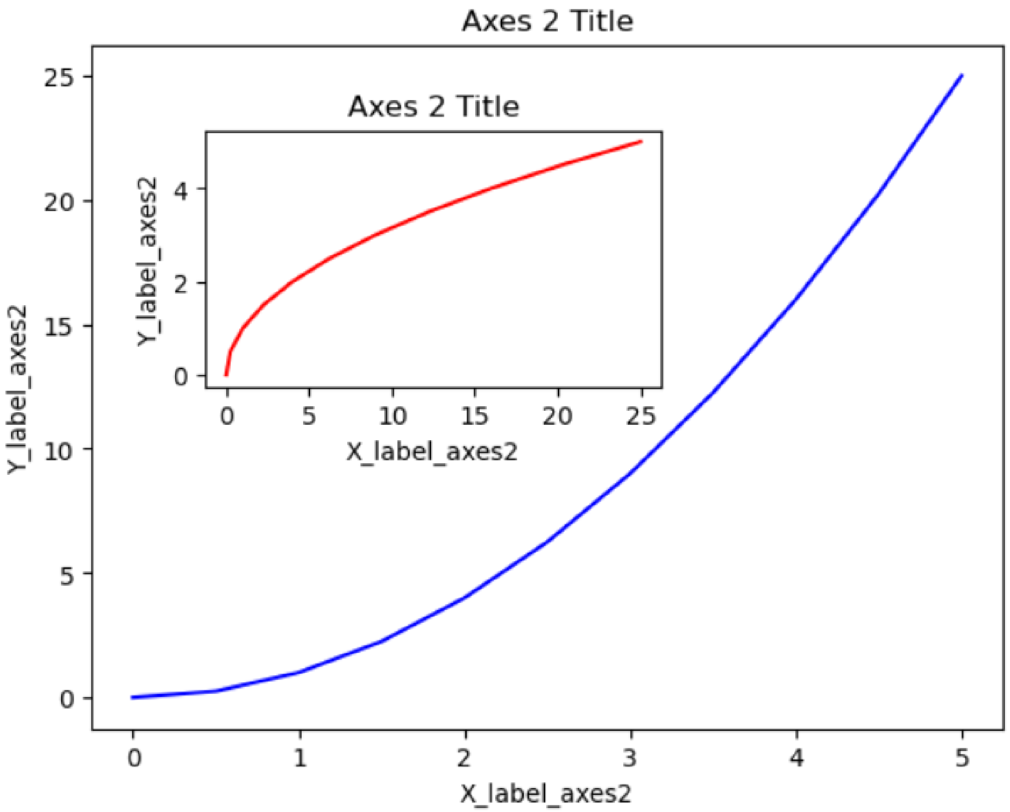






```
[24]: # Creates blank canvas
      fig = plt.figure()
      axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # main axes
      axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3]) # inset axes

      # 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')

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



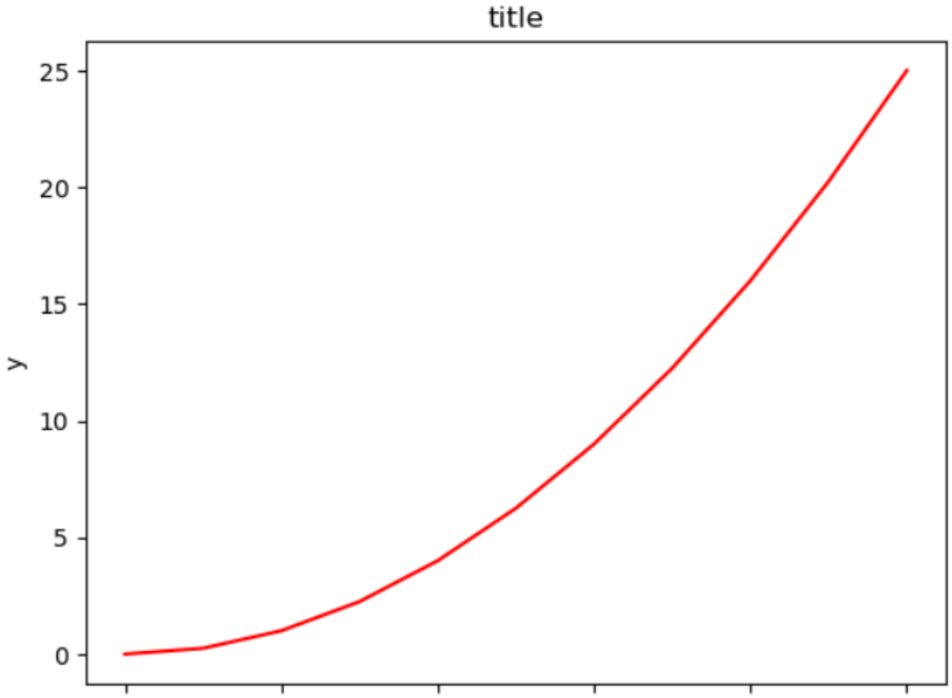


[26]:

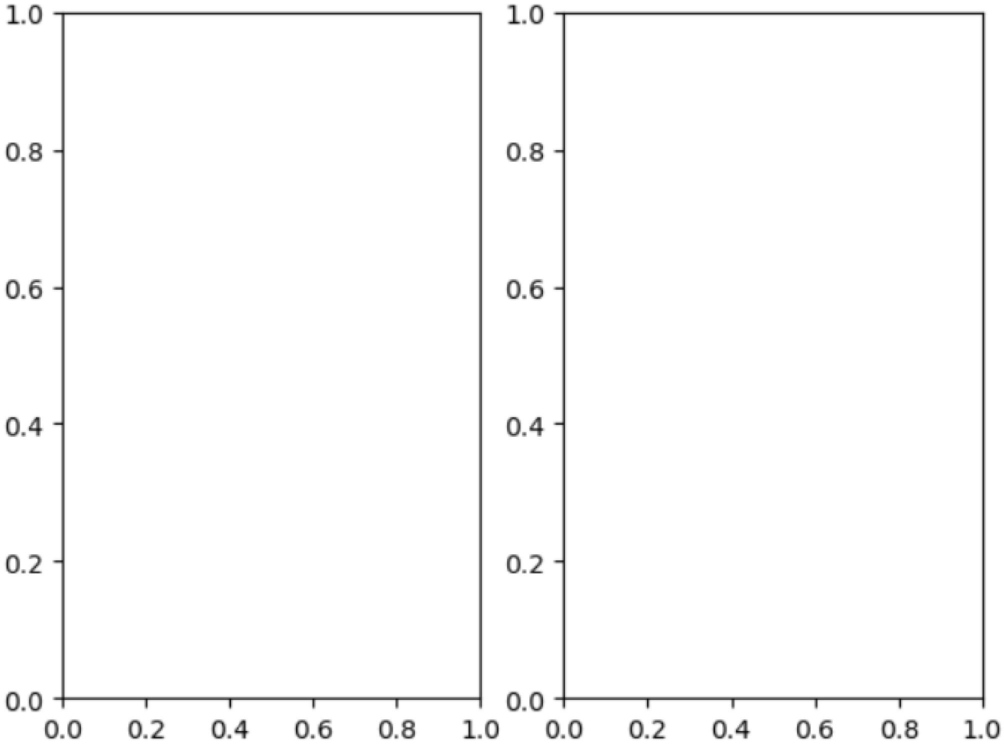
```
# Use similar to plt.figure() except use tuple unpacking to grab fig and ax
fig, axes = plt.subplots()

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

[26]:

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

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



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



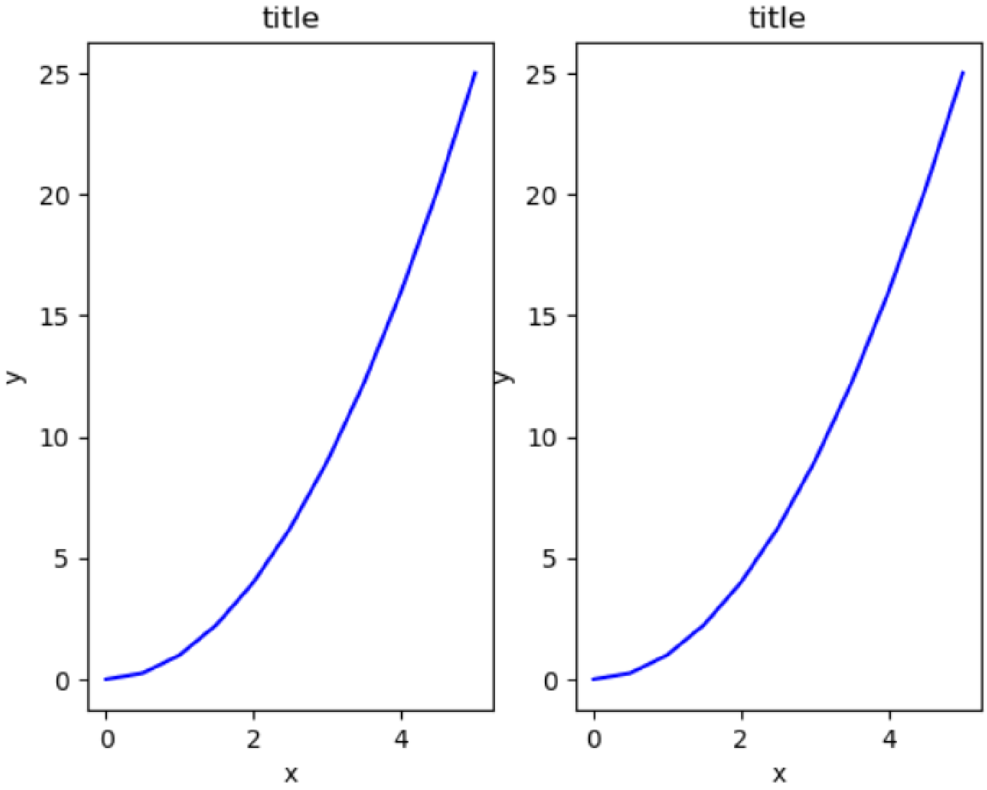
```
[29]: # Axes is an array Of axes to plot on
      axes
```

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

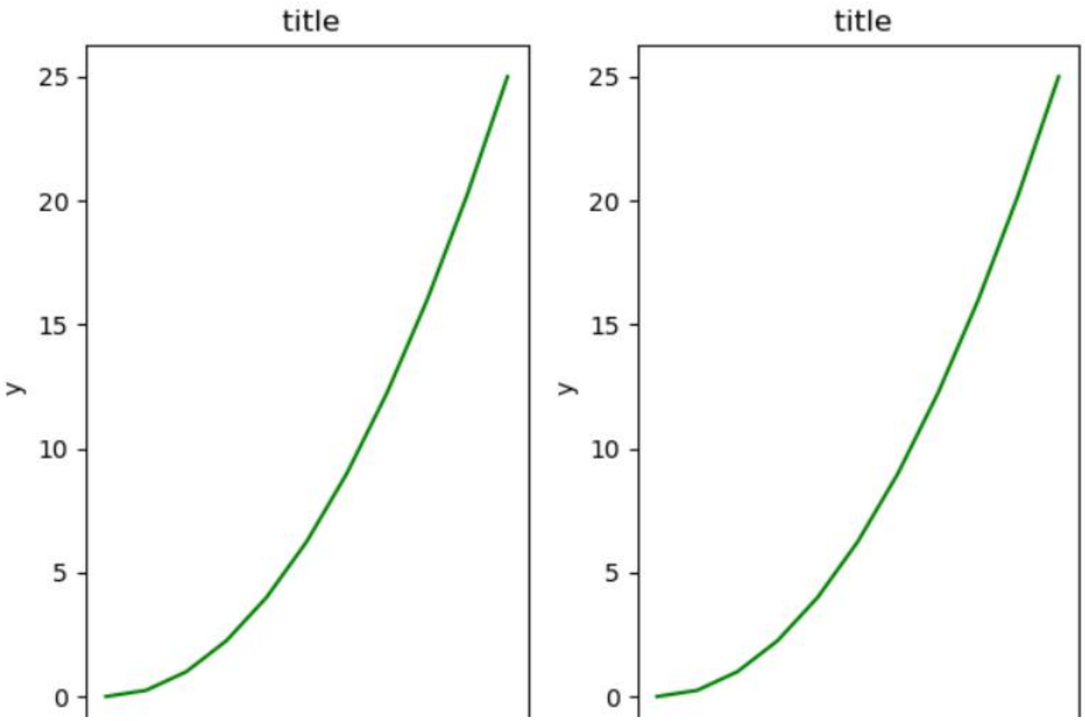
```
[31]: for ax in axes:
          ax.plot(x, y, 'b')
          ax.set_xlabel('x')
          ax.set_ylabel('y')
          ax.set_title('title')
```

```
[32]: #Display the figure object
      fig
```

[32]:



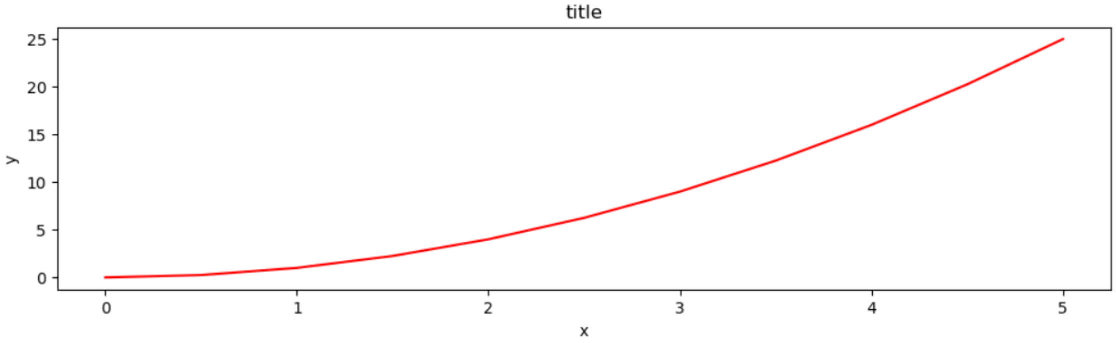
[37]:
```
fig, axes = plt.subplots(nrows=1, ncols=2)

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

fig
plt.tight_layout()
```



[39]:
```
fig = plt.figure(figsize=(8,4), dpi=100)
```

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

[41]:
```
fig, axes = plt.subplots(figsize=(12,3))
axes.plot(x, y, 'r')
axes.set_xlabel('x')
axes.set_ylabel( 'y')
axes.set_title('title');
```



[42]:
```
fig.savefig("filename.png")
```

[44]:
```
fig.savefig("filename.png", dpi=200)
```

```
[42]: fig.savefig("filename.png")
```

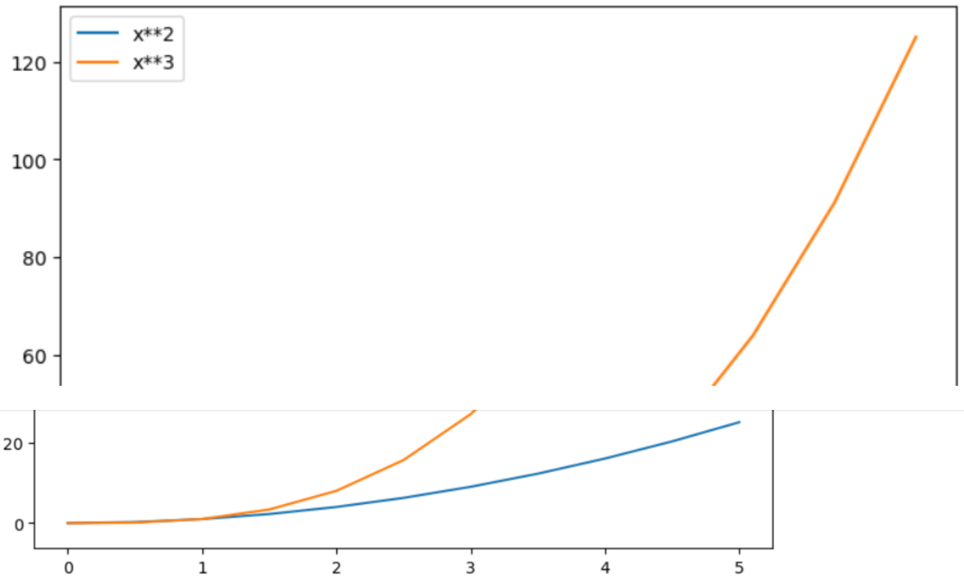
```
[44]: fig.savefig("filename.png", dpi=200)
```

```
[46]: ax.set_title("title");
```

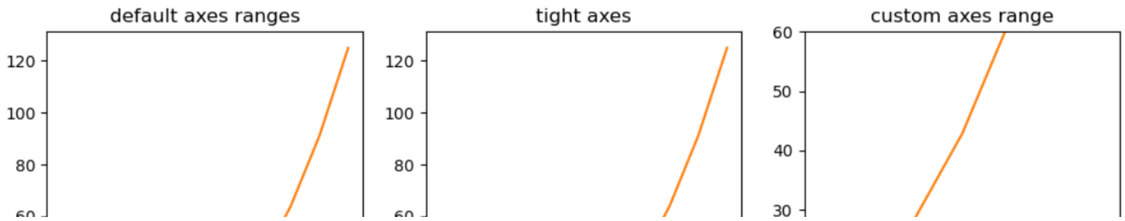
```
[49]: ax.set_xlabel("x")
      ax.set_ylabel("y");
```

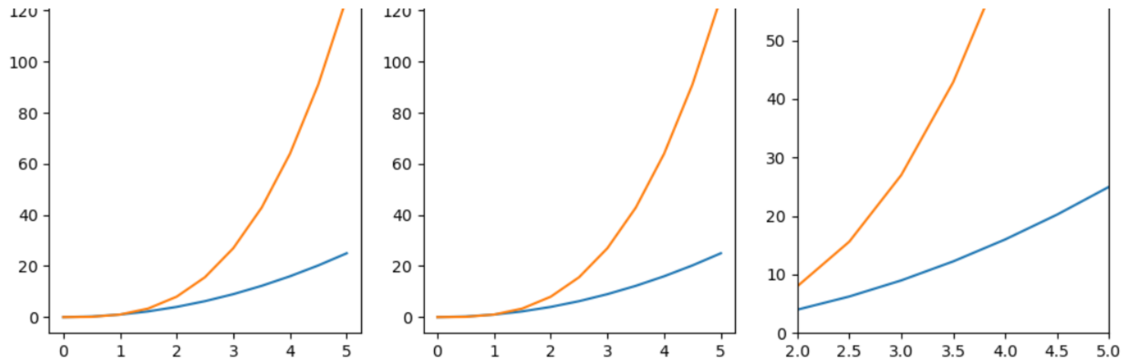
```
[52]: fig = plt.figure()
      ax = fig.add_axes([0,0,1,1])
      ax.plot(x, x**2, label="x**2")
      ax.plot(x, x**3, label="x**3")
      ax.legend()
```

```
[52]: <matplotlib.legend.Legend at 0x1f5a3dd22e0>
```



```
[55]: 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");
```





```
[58]: import pandas as pd
      import matplotlib.pyplot as plt
      import matplotlib.image as mpimg
      import seaborn as sns
      %matplotlib inline

      sns.get_dataset_names()
```

```
[58]: ['anagrams',
       'anscombe',
       'attention',
       'brain_networks',
       'car_crashes',
       'diamonds',
       'dots',
```

[58]:

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

[60]:

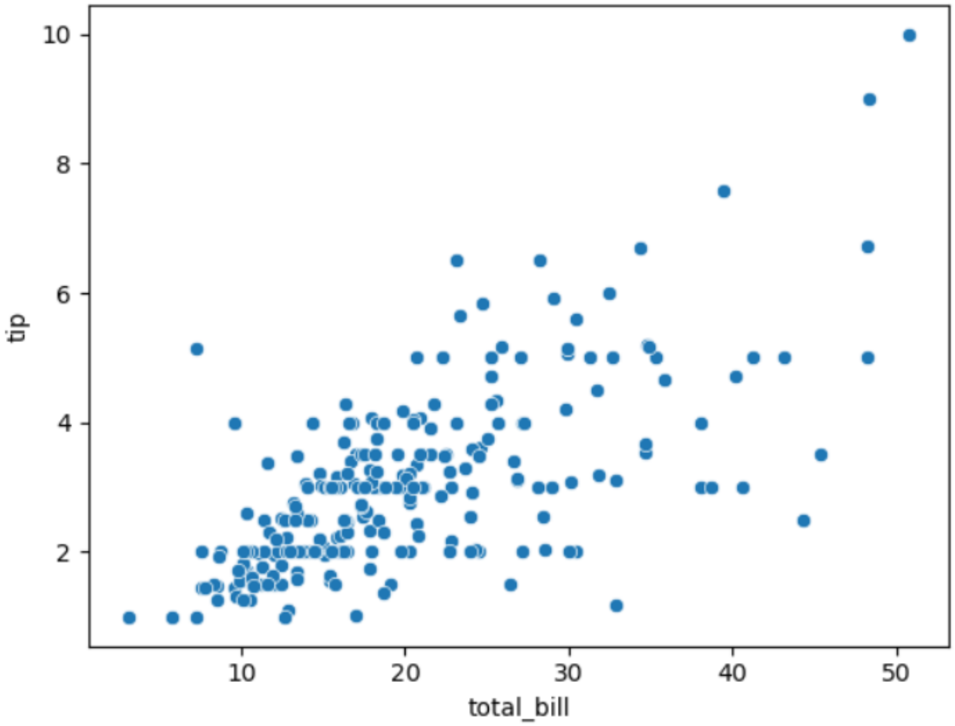
```
tips = sns.load_dataset("tips")
tips.head()
```

[60]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2

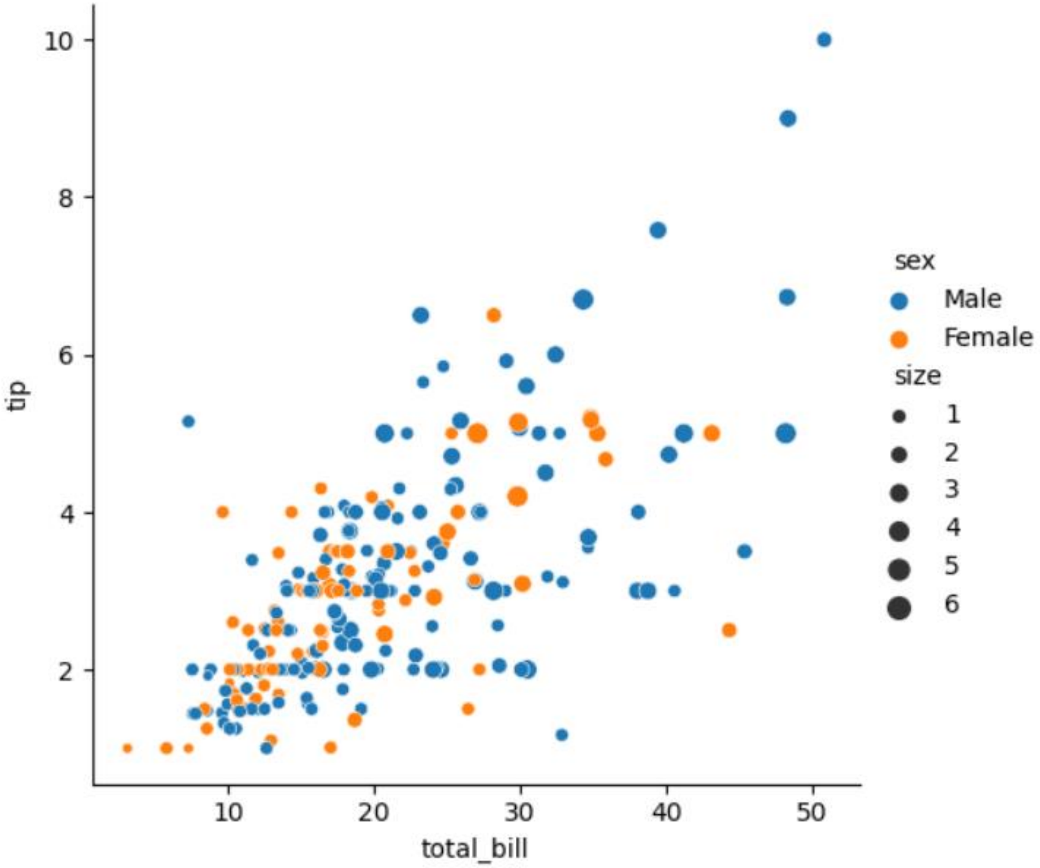
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

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



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

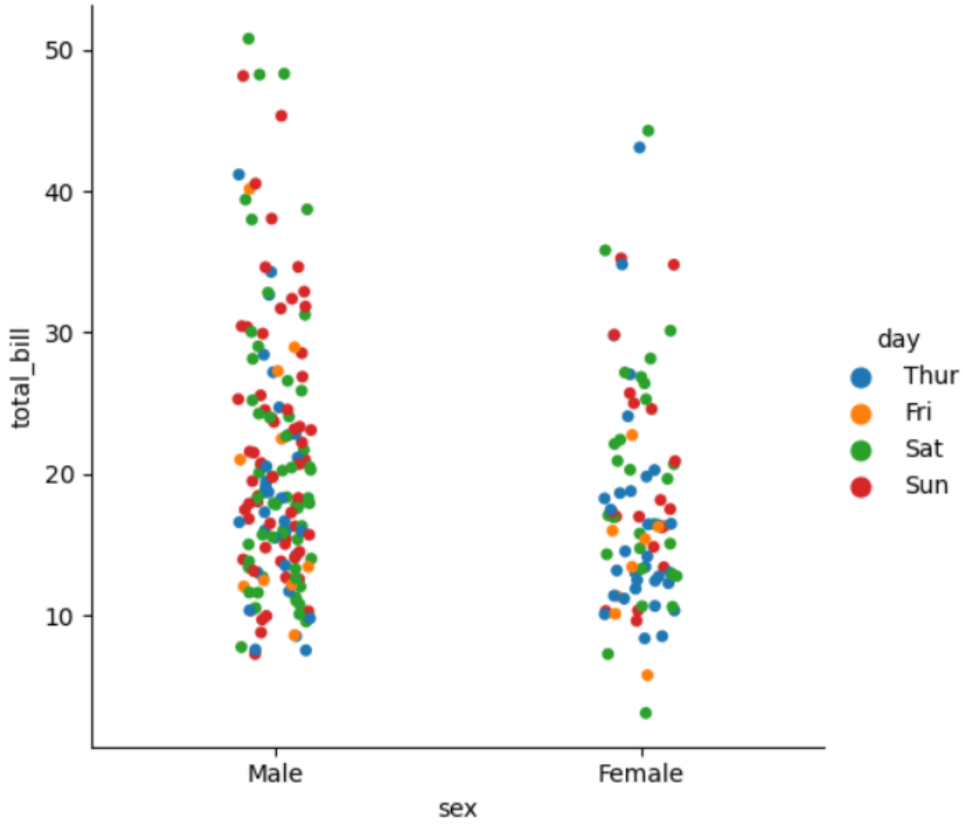
```
[66]: <seaborn.axisgrid.FacetGrid at 0x1f5a7faf760>
```



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

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

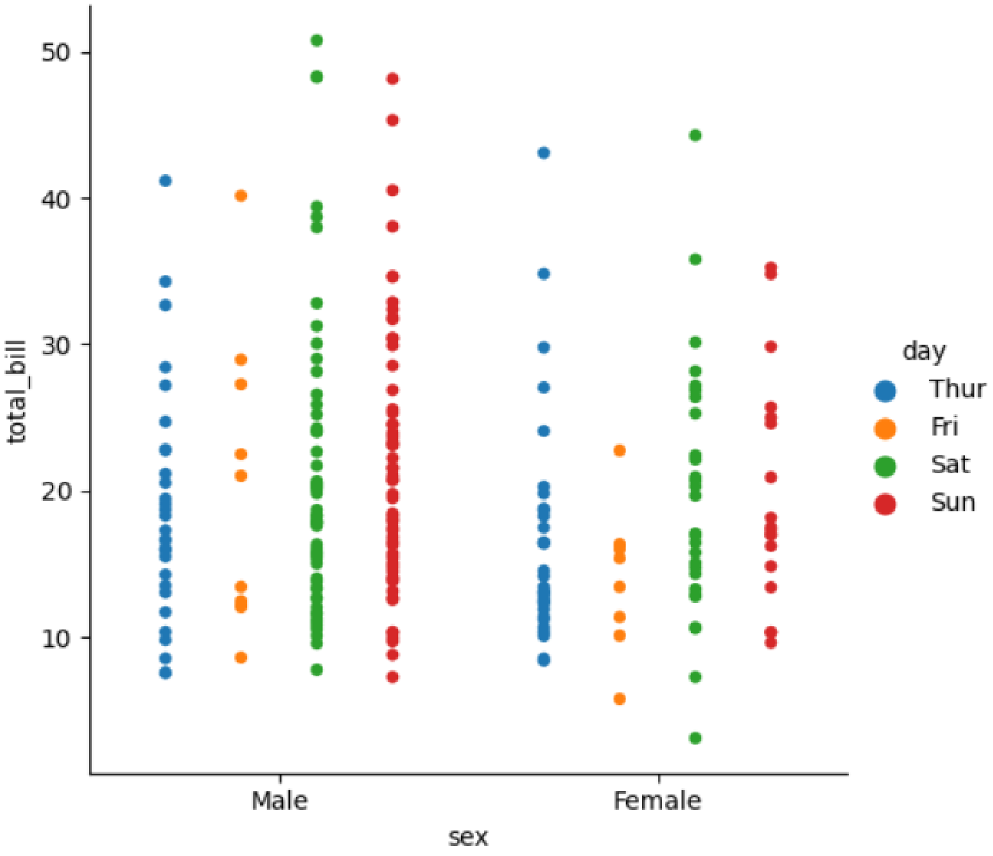
```
[68]: <seaborn.axisgrid.FacetGrid at 0x1f5a7faf340>
```



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

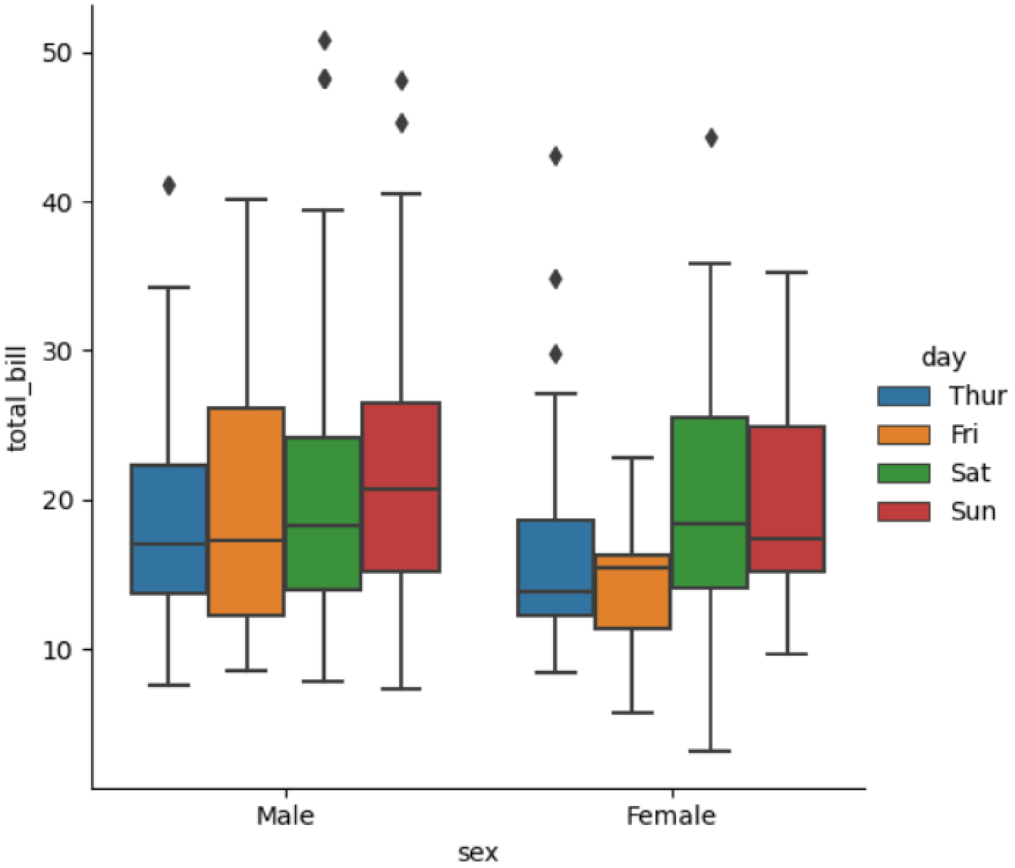
```
[70]: sns.catplot(x="sex", y="total_bill", hue="day", data=tips, kind="strip",
                  jitter=False, dodge=True)
```

```
[70]: <seaborn.axisgrid.FacetGrid at 0x1f5a4162460>
```



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

```
[72]: <seaborn.axisgrid.FacetGrid at 0x1f5a80e5e50>
```



```
[ ]: %matplotlib inline
```

```
[ ]: import pandas as pd
```

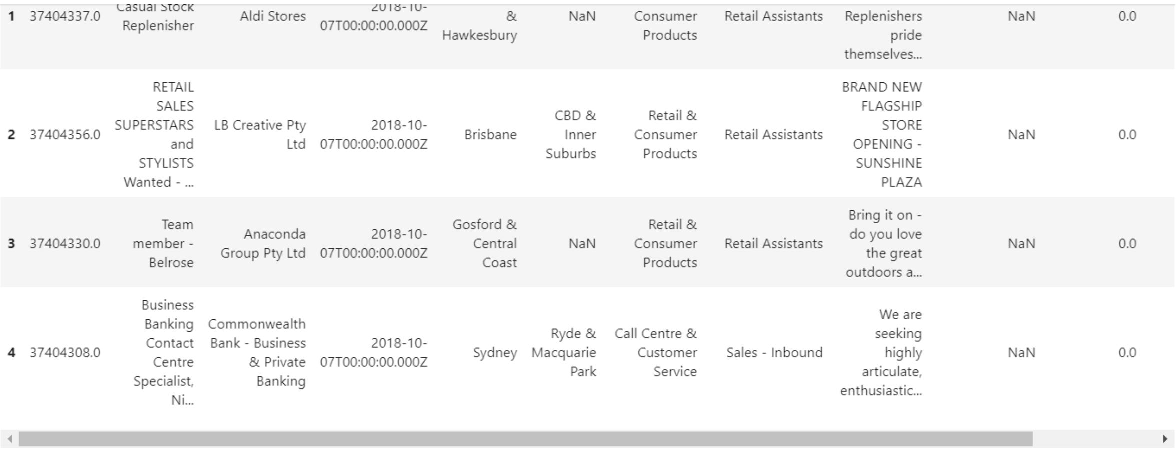
```
[ ]: import matplotlib.pyplot as plt
```

```
[85]: df = pd.read_csv('job-market.csv')
```

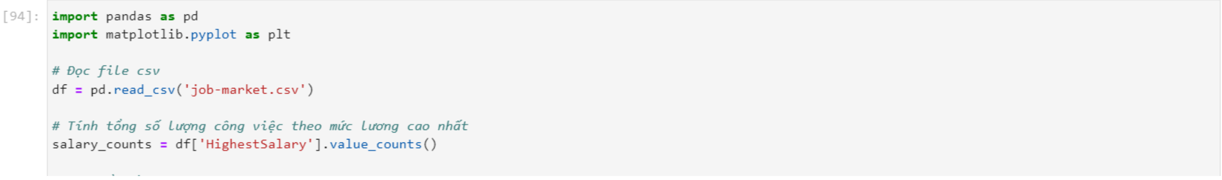
```
[86]: df.head()
```

[86]:

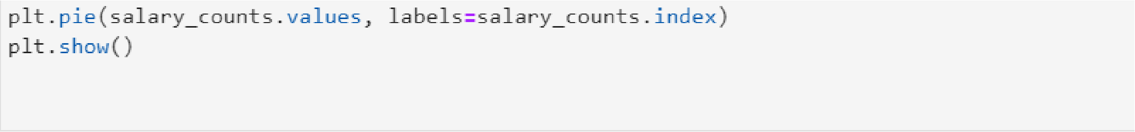
	Id	Title	Company	Date	Location	Area	Classification	SubClassification	Requirement	FullDescription	LowestSalary	Hig
0	37404348.0	Casual Stock Replenisher	Aldi Stores	2018-10-07T00:00:00.000Z	Sydney	North West & Hills District	Retail & Consumer Products	Retail Assistants	Our Casual Stock Replenishers pride themselves...	NaN	0.0	
1	37404337.0	Casual Stock Replenisher	Aldi Stores	2018-10-07T00:00:00.000Z	Richmond & Hawkesbury	NaN	Retail & Consumer Products	Retail Assistants	Our Casual Stock Replenishers pride themselves...	NaN	0.0	
		RETAIL							BRAND NEW			



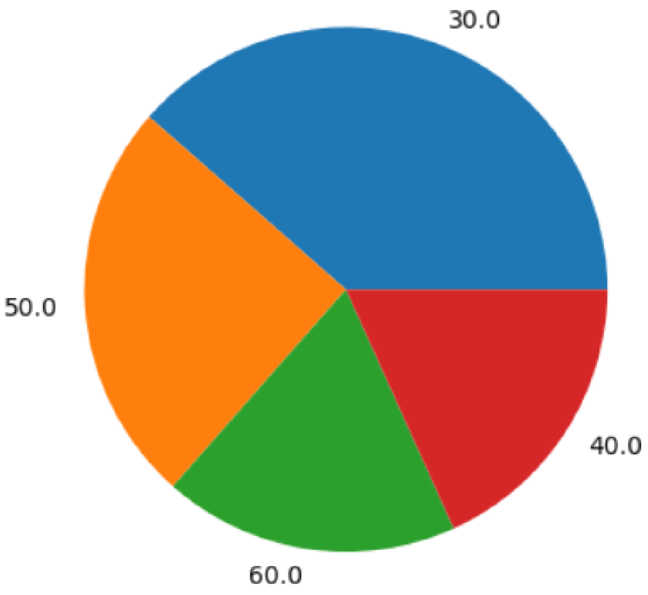
1	37404337.0	Casual Stock Replenisher	Aldi Stores	2018-10-07T00:00:00.000Z	& Hawkesbury	NaN	Consumer Products	Retail Assistants	Replenishers pride themselves...	NaN	0.0
2	37404356.0	RETAIL SALES SUPERSTARS and STYLISTS Wanted - ...	LB Creative Pty Ltd	2018-10-07T00:00:00.000Z	Brisbane	CBD & Inner Suburbs	Retail & Consumer Products	Retail Assistants	BRAND NEW FLAGSHIP STORE OPENING - SUNSHINE PLAZA	NaN	0.0
3	37404330.0	Team member - Belrose	Anaconda Group Pty Ltd	2018-10-07T00:00:00.000Z	Gosford & Central Coast	NaN	Retail & Consumer Products	Retail Assistants	Bring it on - do you love the great outdoors a...	NaN	0.0
4	37404308.0	Business Banking Contact Centre Specialist, Ni...	Commonwealth Bank - Business & Private Banking	2018-10-07T00:00:00.000Z	Sydney	Ryde & Macquarie Park	Call Centre & Customer Service	Sales - Inbound	We are seeking highly articulate, enthusiastic...	NaN	0.0

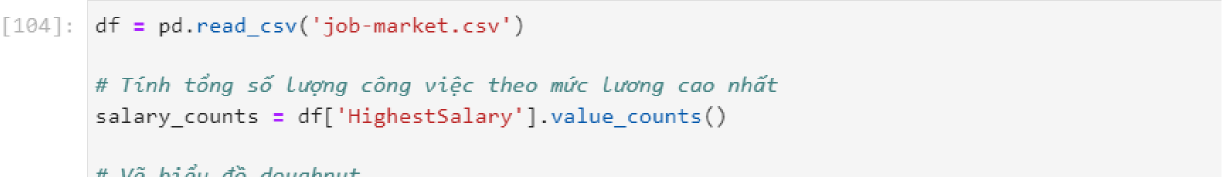


```
[94]: import pandas as pd
      import matplotlib.pyplot as plt

      # Đọc file csv
      df = pd.read_csv('job-market.csv')

      # Tính tổng số lượng công việc theo mức lương cao nhất
      salary_counts = df['HighestSalary'].value_counts()
```

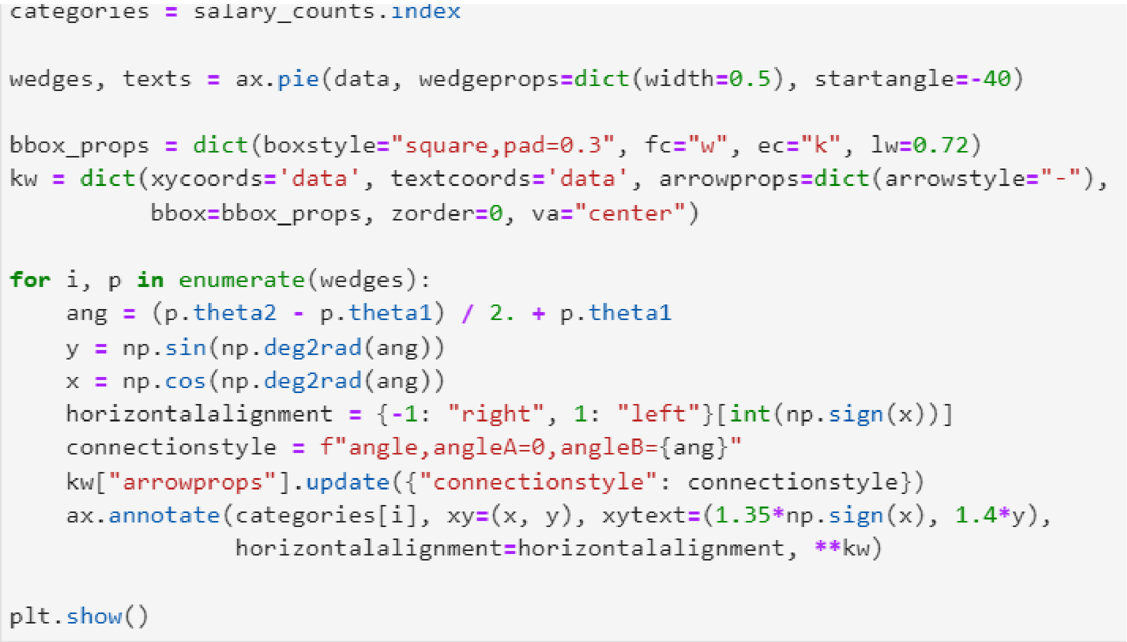


```
plt.pie(salary_counts.values, labels=salary_counts.index)
plt.show()
```

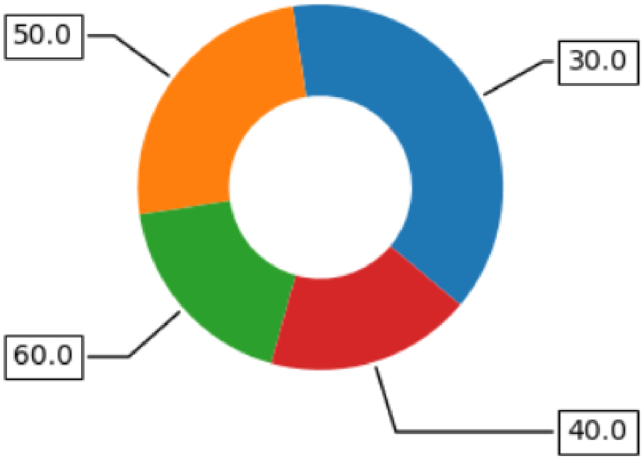




```
[104]: df = pd.read_csv('job-market.csv')

       # Tính tổng số lượng công việc theo mức lương cao nhất
       salary_counts = df['HighestSalary'].value_counts()

       # Vẽ biểu đồ doughnut
```



```python
categories = salary_counts.index

wedges, texts = ax.pie(data, wedgeprops=dict(width=0.5), startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(xycoords='data', textcoords='data', arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1) / 2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = f"angle,angleA=0,angleB={ang}"
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(categories[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)

plt.show()
```


50.0
30.0
60.0
40.0