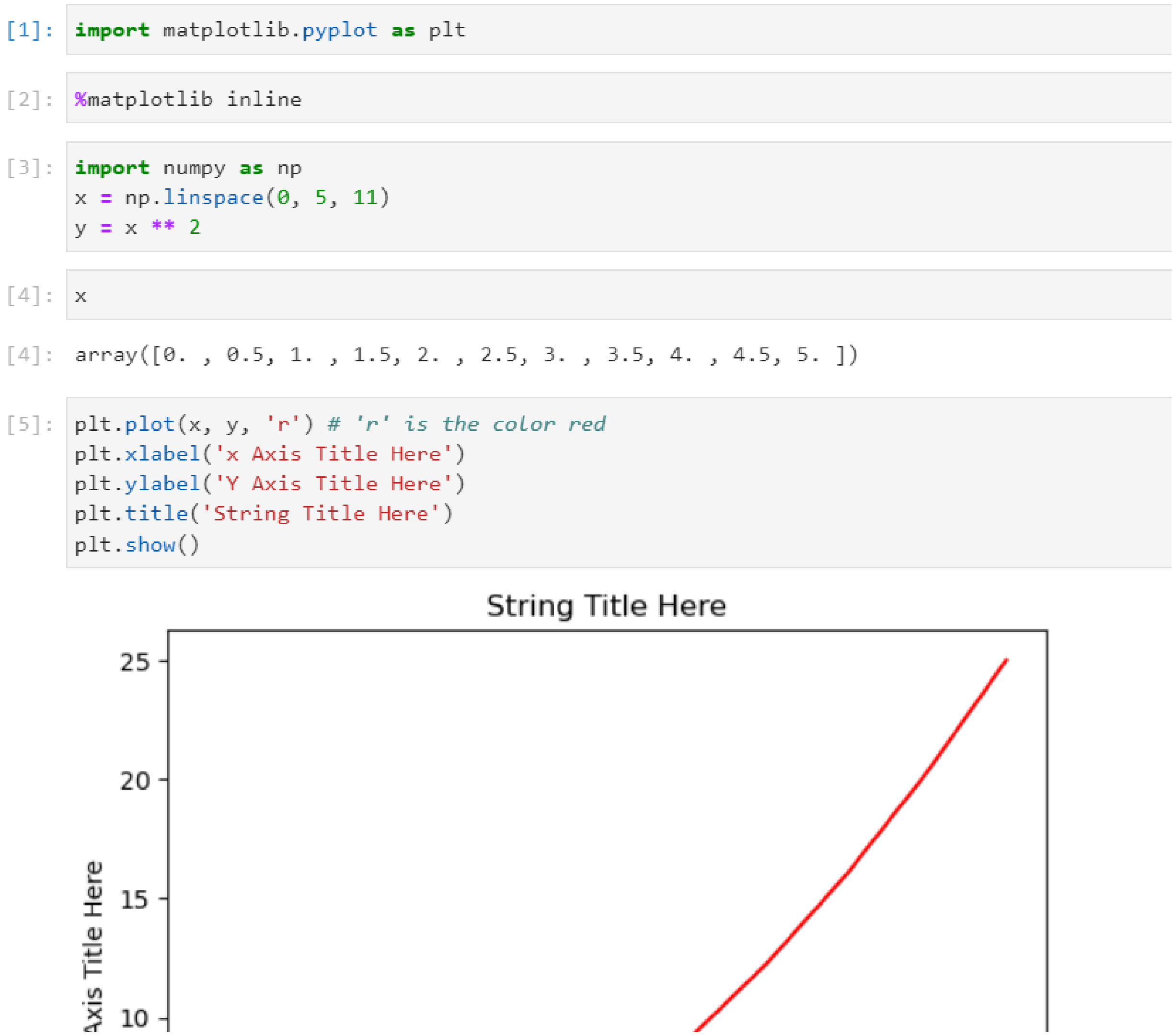
Name: Nguyễn Thanh Lộc

ID: 19521763

Lab3

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



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

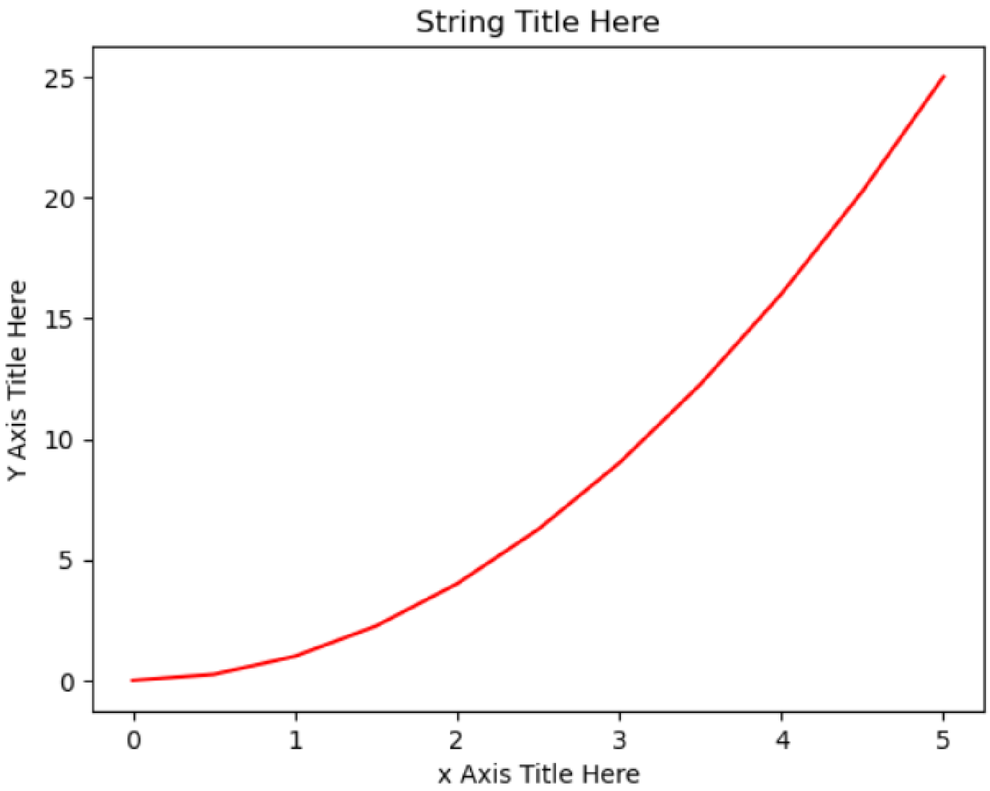
```
[2]: %matplotlib inline
```

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

```
[4]: x
```

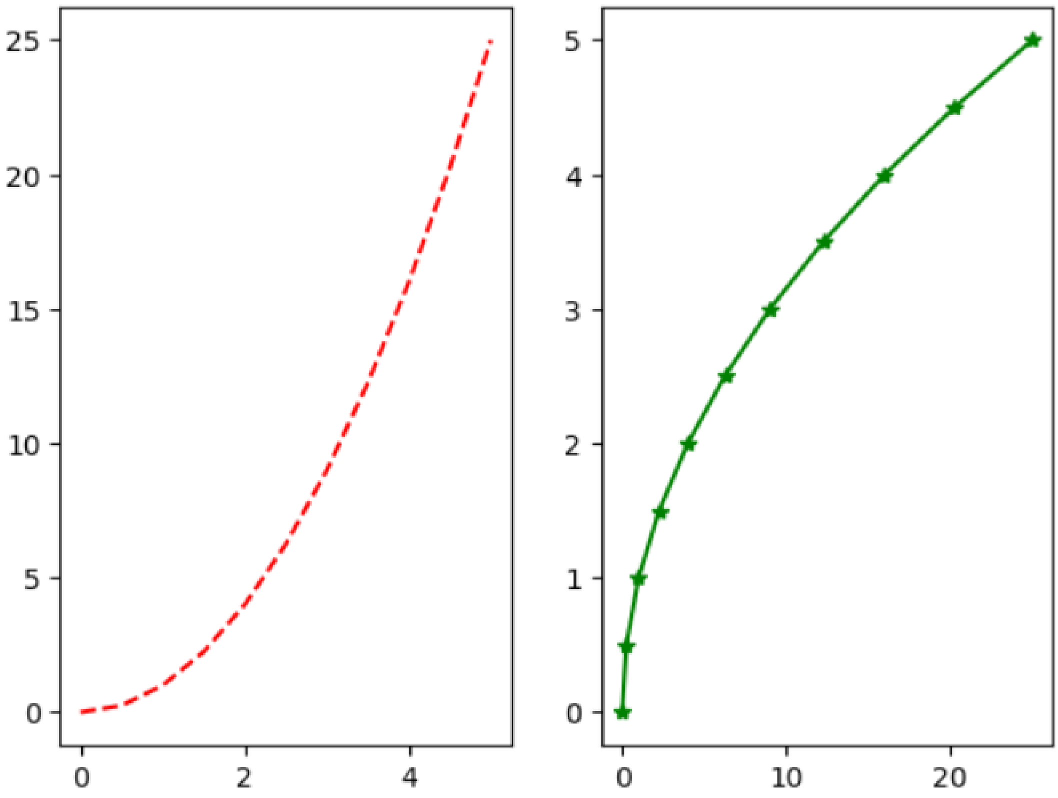
```
[4]: array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

```
[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()
```



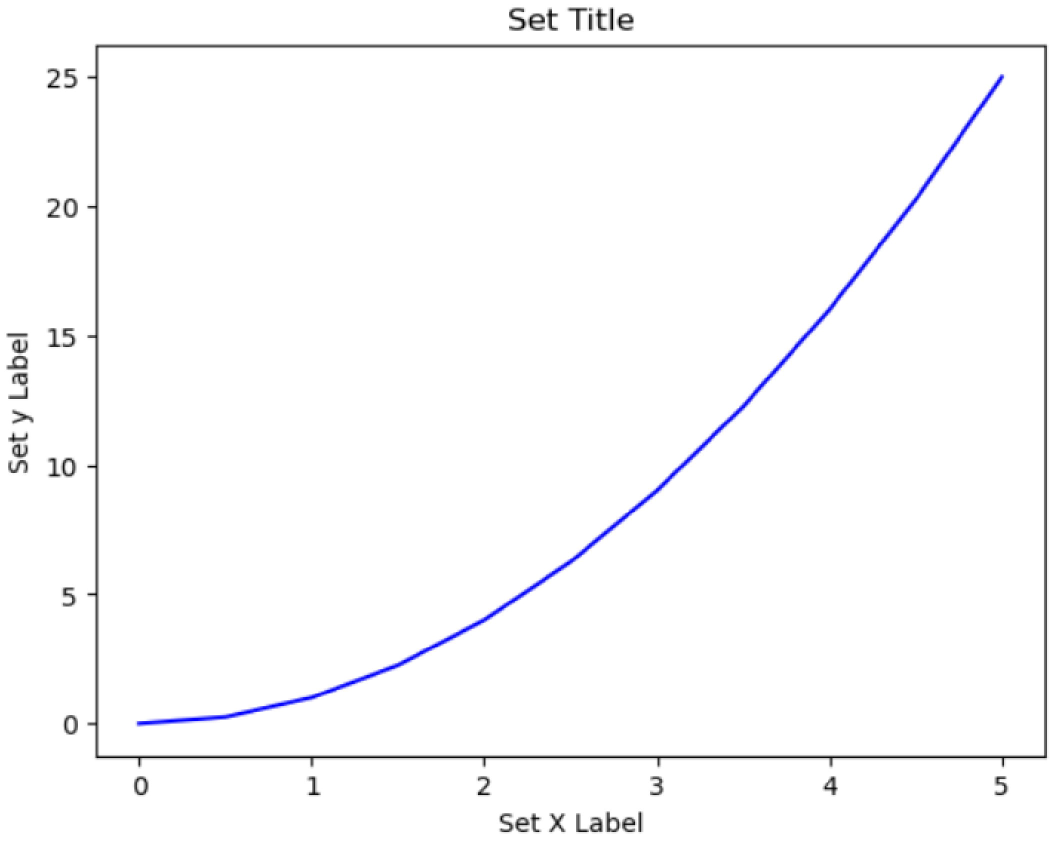
[8]:

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

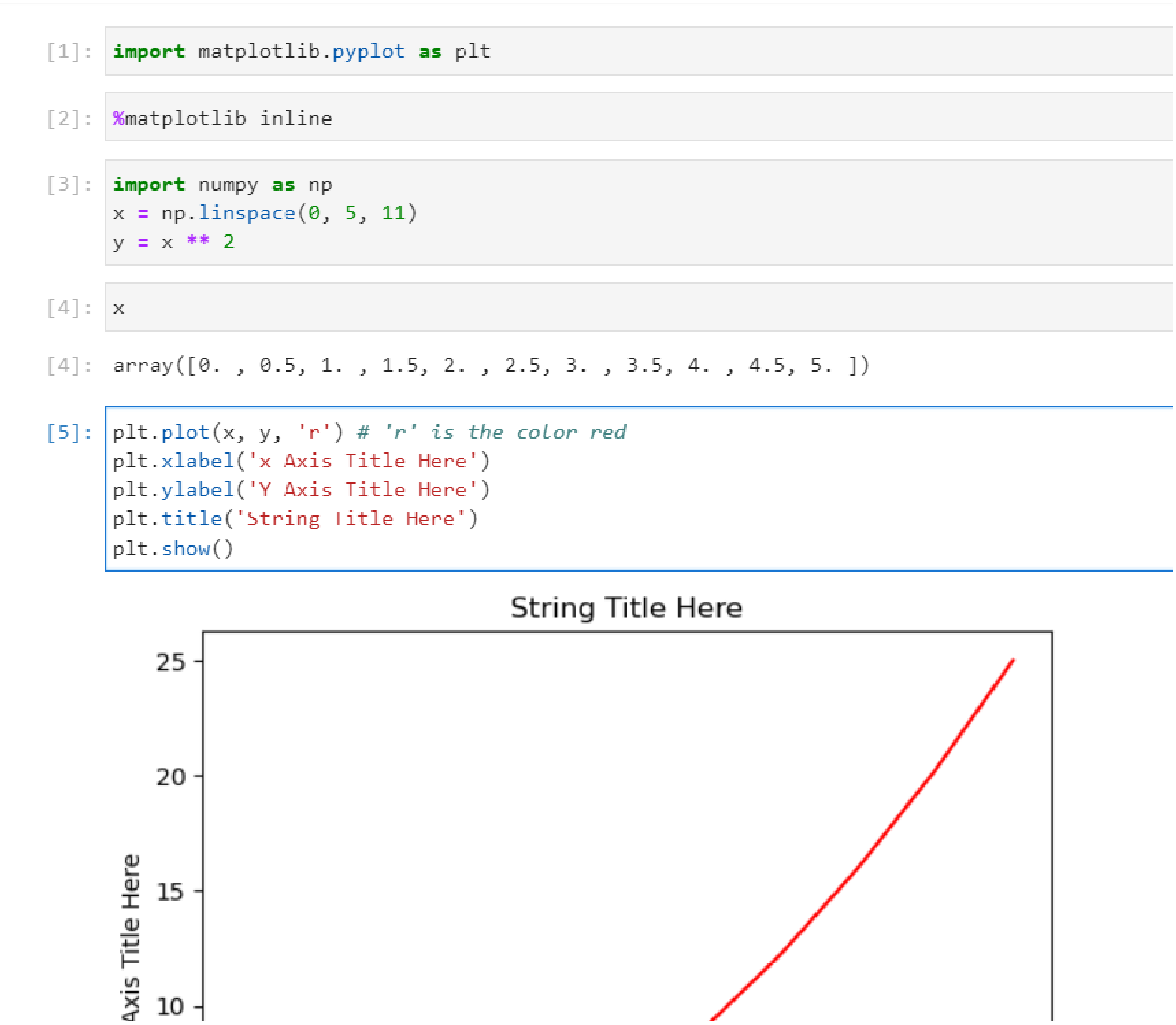


```
[11]: # Create Figure (empty canvas)
      fig = plt.figure()
      # Add set of axes to figure
      axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height
      axes.plot(x, y, 'b')
      axes.set_xlabel('Set X Label') # Notice the use Of set to begin methods
      axes.set_ylabel('Set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

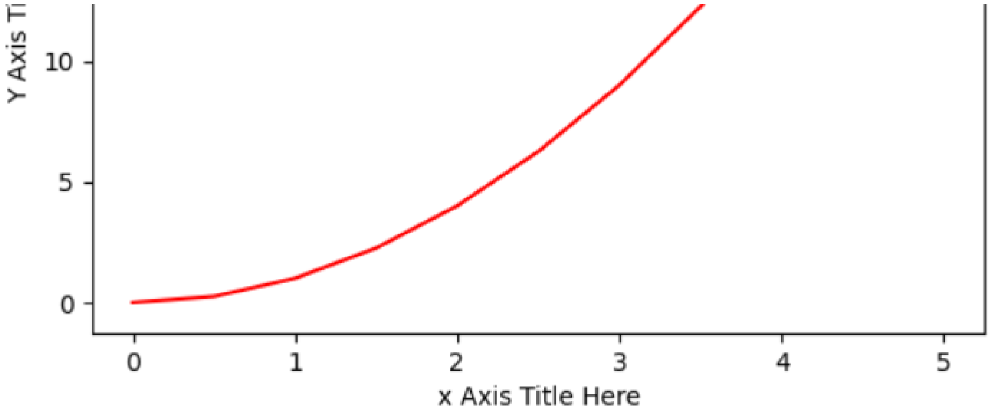
```
[2]: %matplotlib inline
```

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

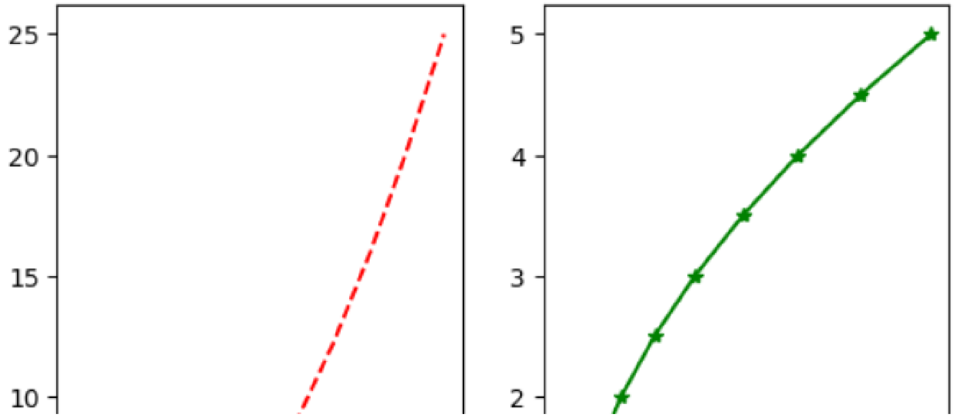
```
[4]: x
```

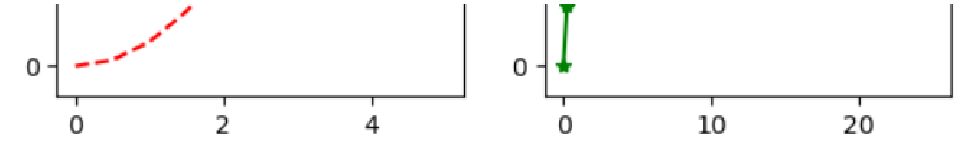
```
[4]: array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

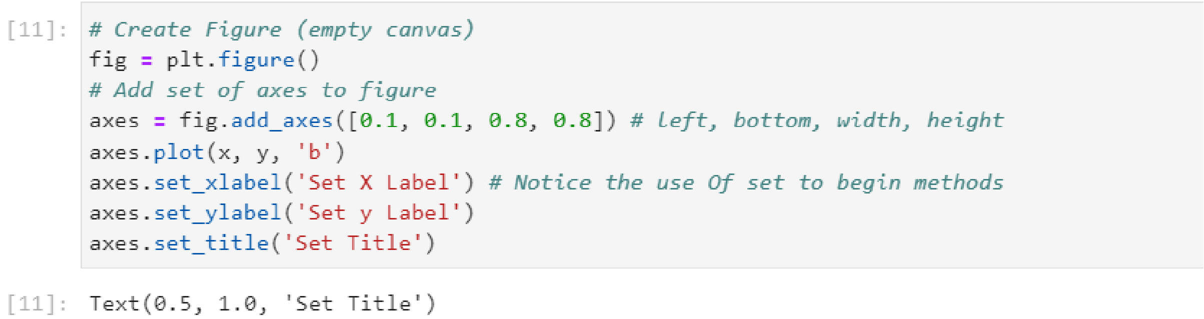
```
[5]: plt.plot(x, y, 'r') # 'r' is the color red
     plt.xlabel('x Axis Title Here')
     plt.ylabel('Y Axis Title Here')
     plt.title('String Title Here')
     plt.show()
```



```
[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*-');
```

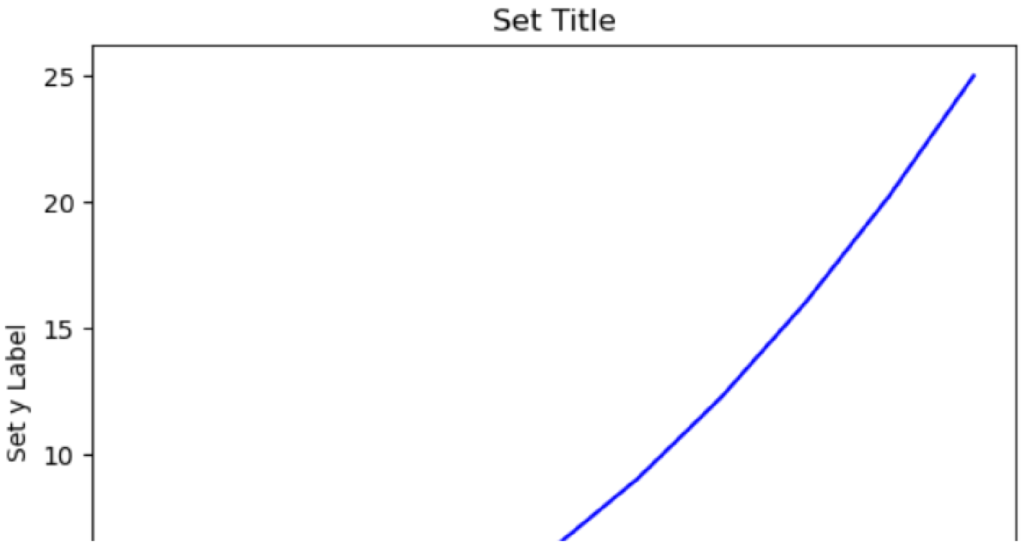


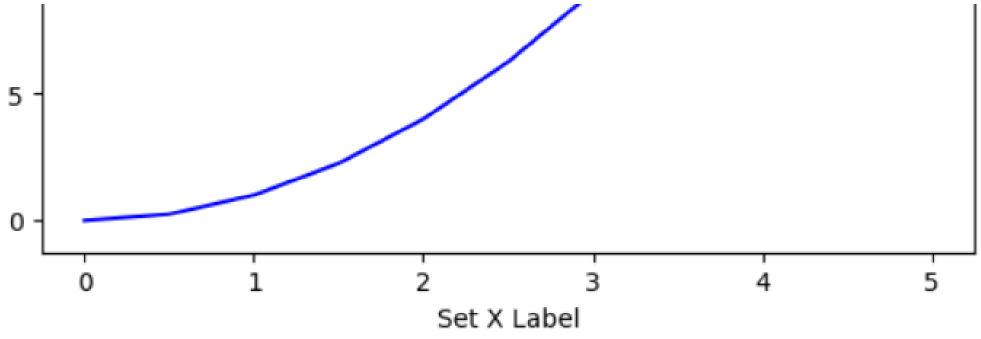


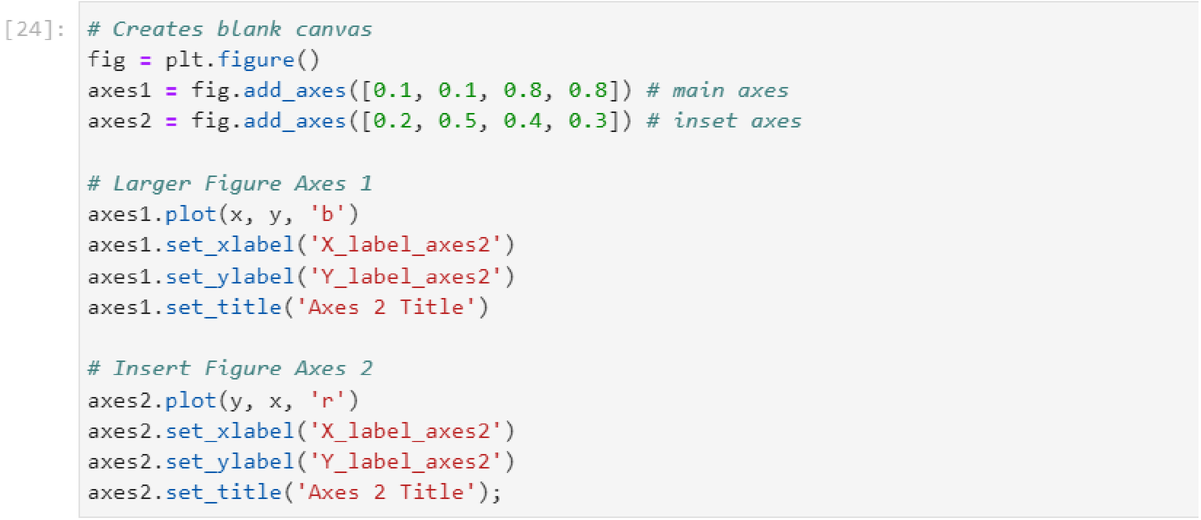


```
[11]: # Create Figure (empty canvas)
      fig = plt.figure()
      # Add set of axes to figure
      axes = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # left, bottom, width, height
      axes.plot(x, y, 'b')
      axes.set_xlabel('Set X Label') # Notice the use Of set to begin methods
      axes.set_ylabel('Set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

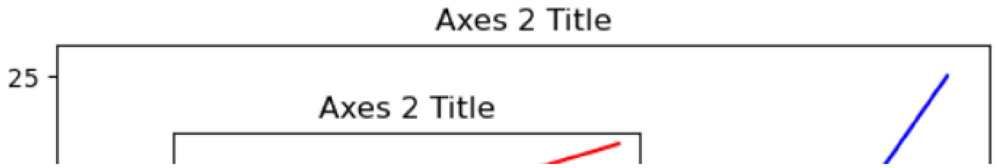
```
[11]: Text(0.5, 1.0, 'Set Title')
```

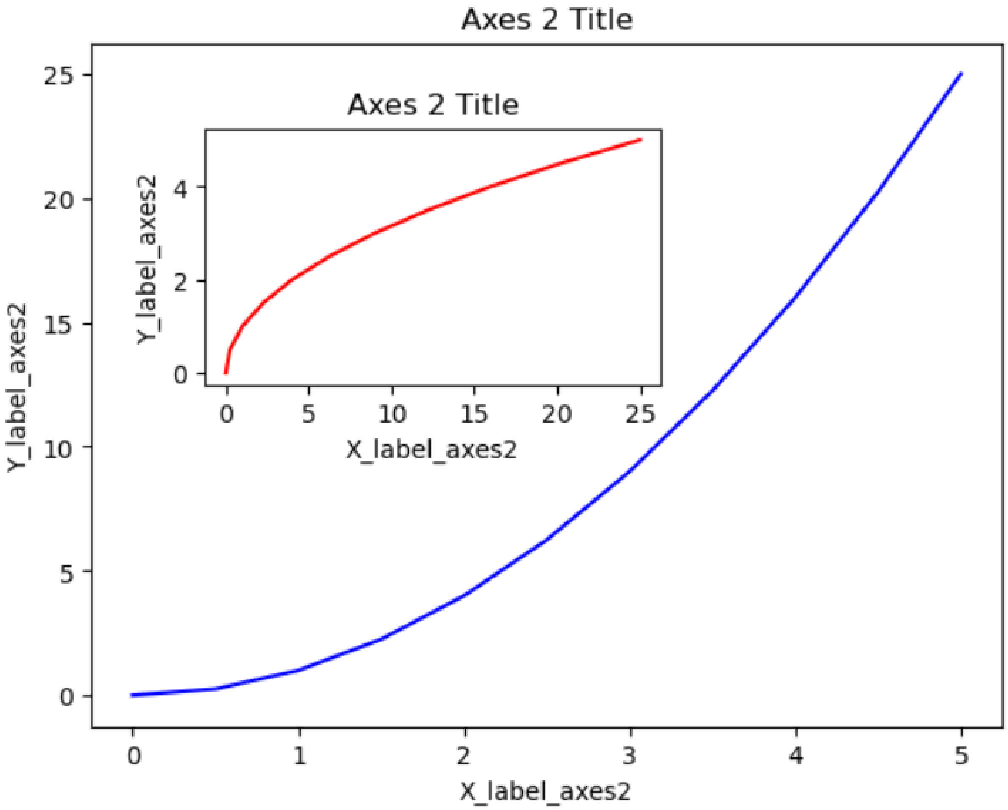






```
# Creates blank canvas
fig = plt.figure()
axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8]) # main axes
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3]) # inset axes

# 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');
```

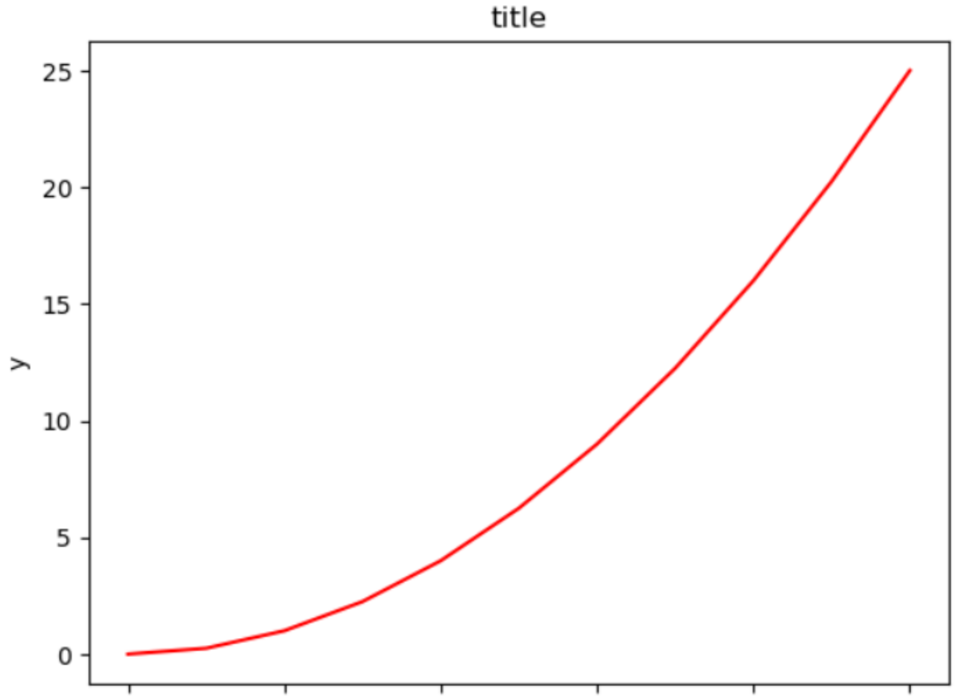




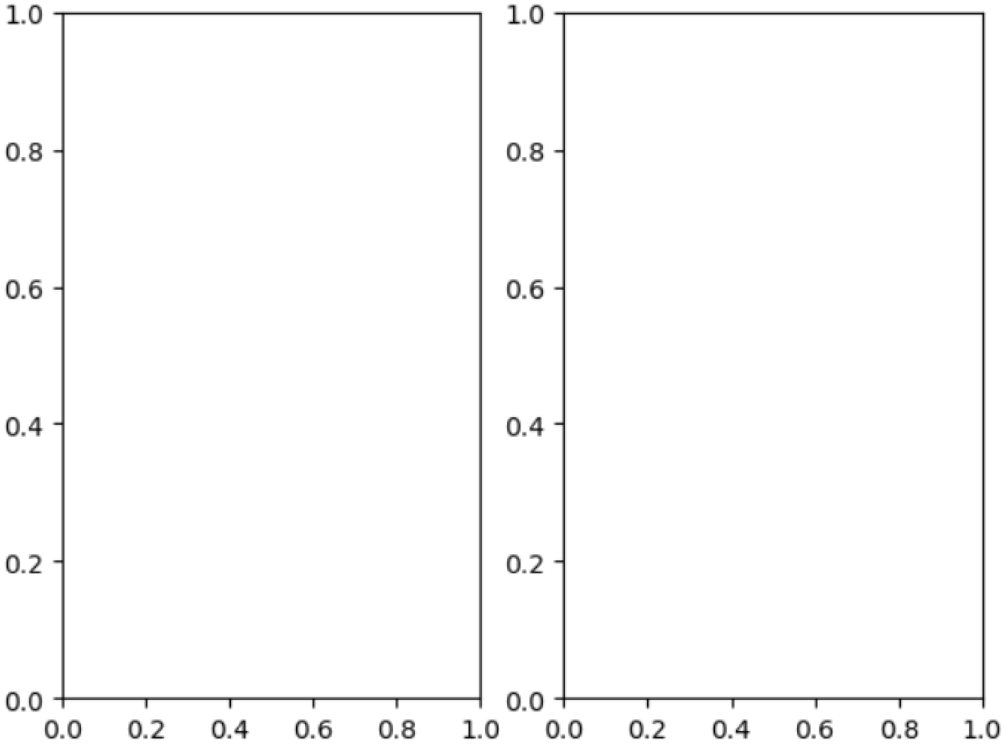
```
[26]: # Use similar to plt.figure() except use tuple unpacking to grab fig and ax
      fig, axes = plt.subplots()

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

[26]:

```
# 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)
```

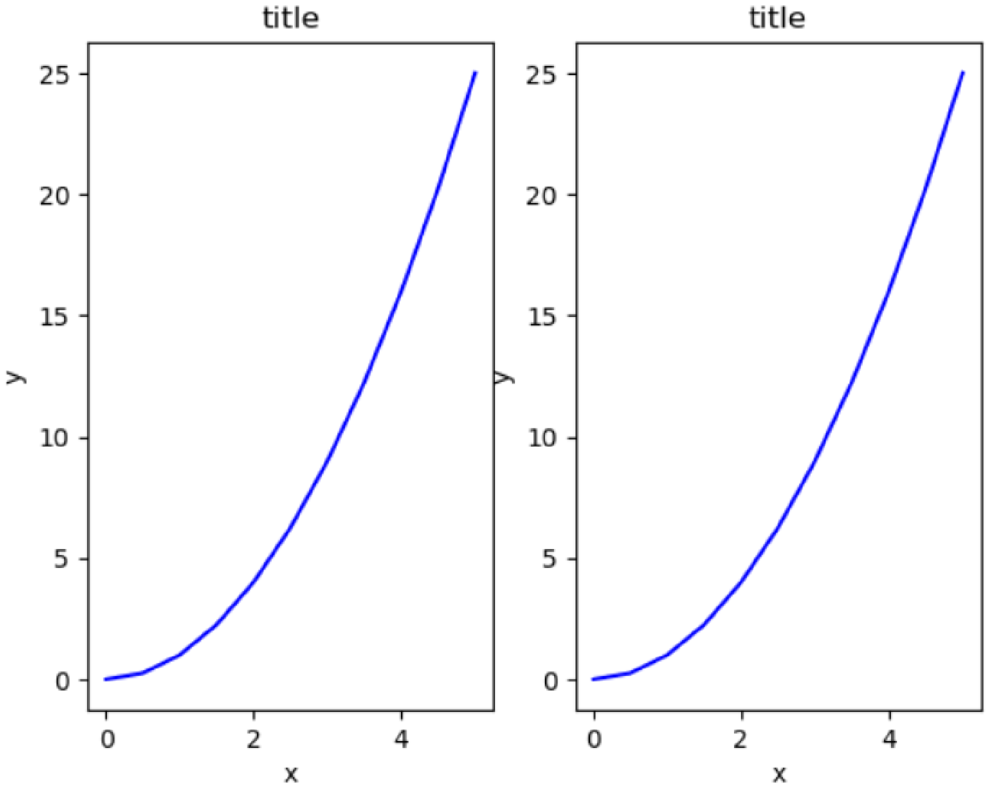
[31]:

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

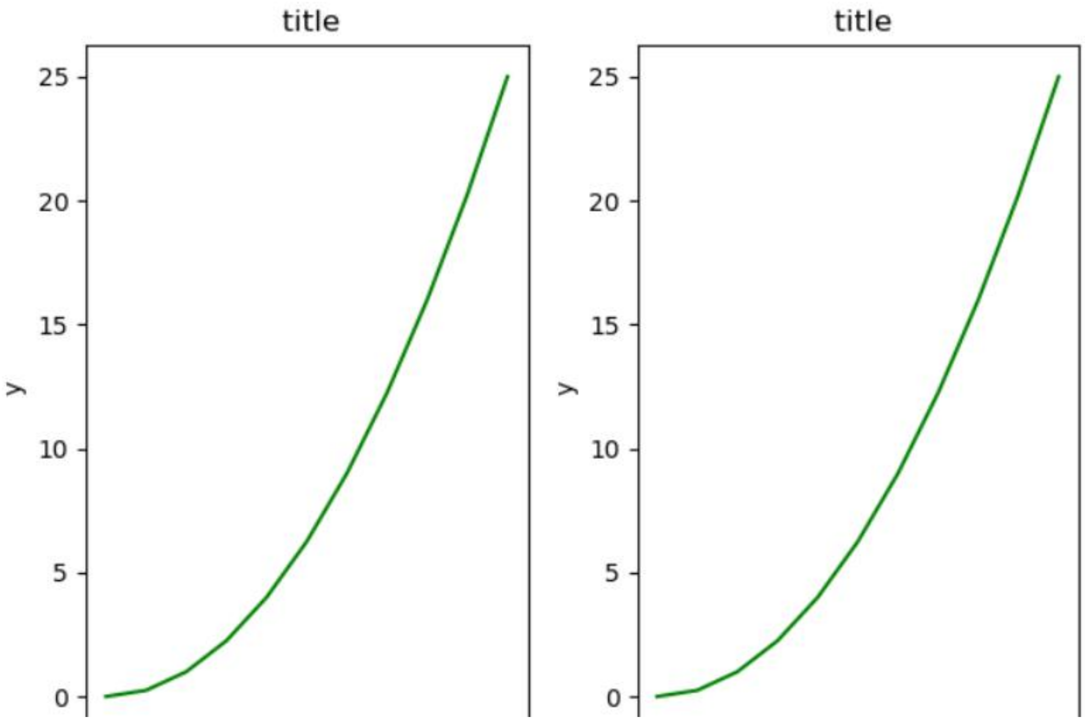
[32]:

```
#Display the figure object
fig
```

[32]:



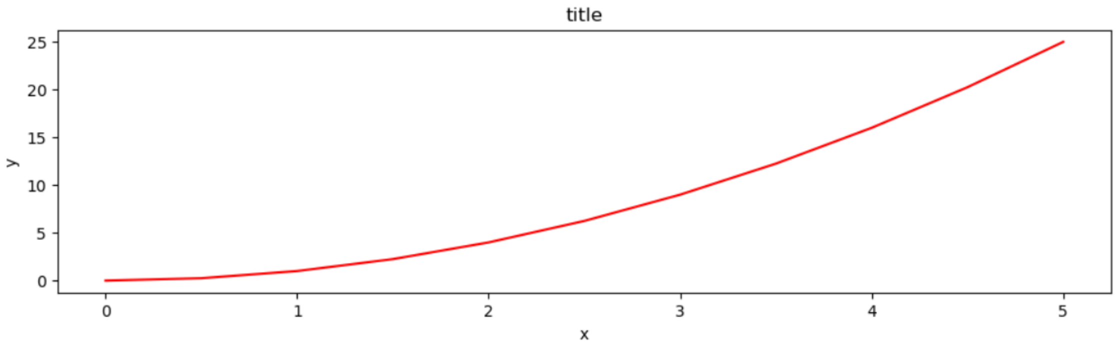
```
[37]: fig, axes = plt.subplots(nrows=1, ncols=2)

      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")
```

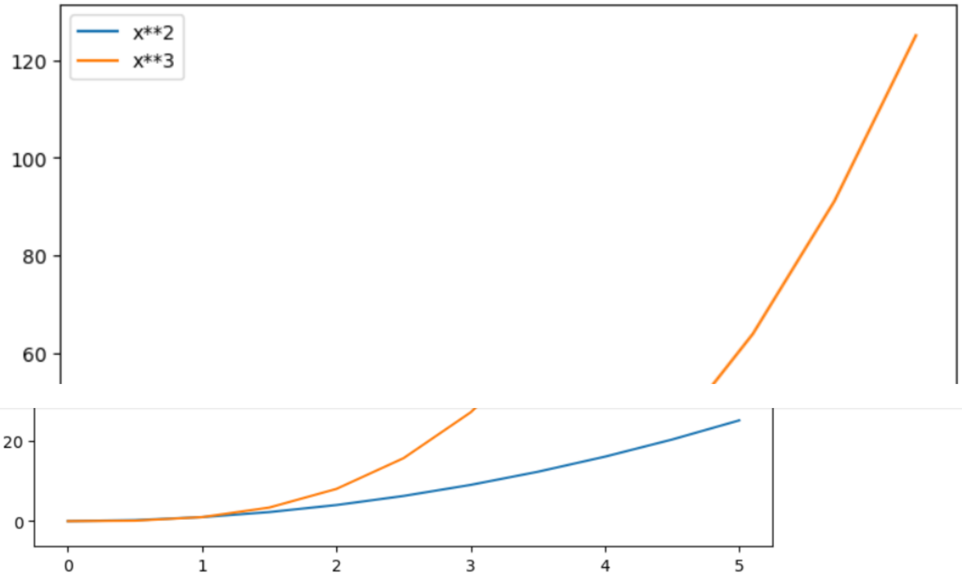
```
[44]: fig.savefig("filename.png", dpi=200)
```

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

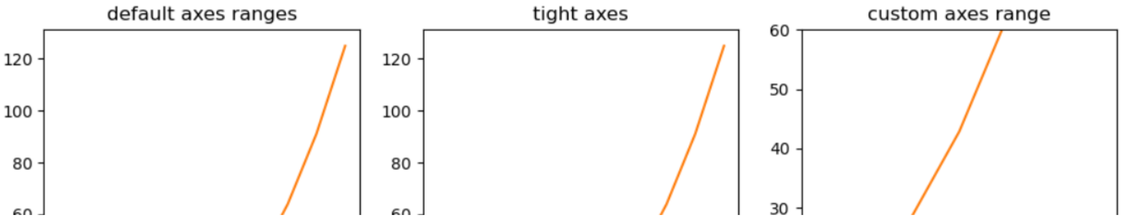
```
[49]: ax.set_xlabel("x")
      ax.set_ylabel("y");
```

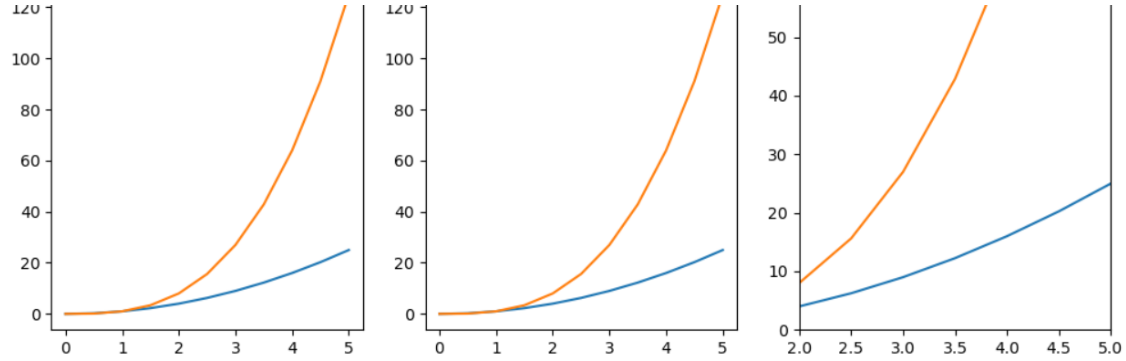
```
[52]: fig = plt.figure()
      ax = fig.add_axes([0,0,1,1])
      ax.plot(x, x**2, label="x**2")
      ax.plot(x, x**3, label="x**3")
      ax.legend()
```

```
[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()
```

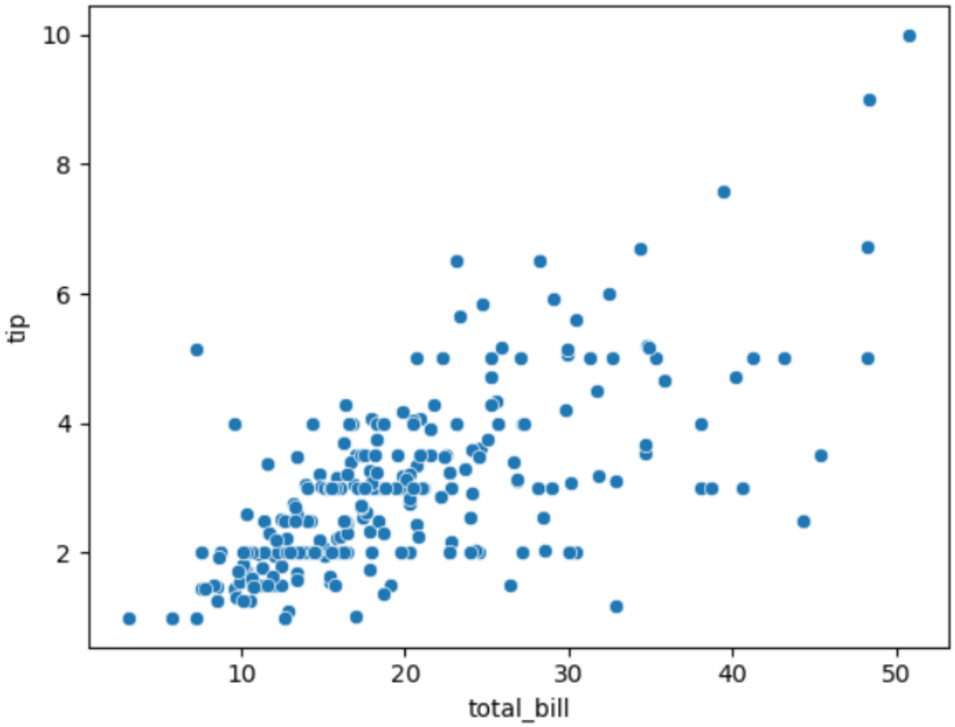
[60]:

	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]:

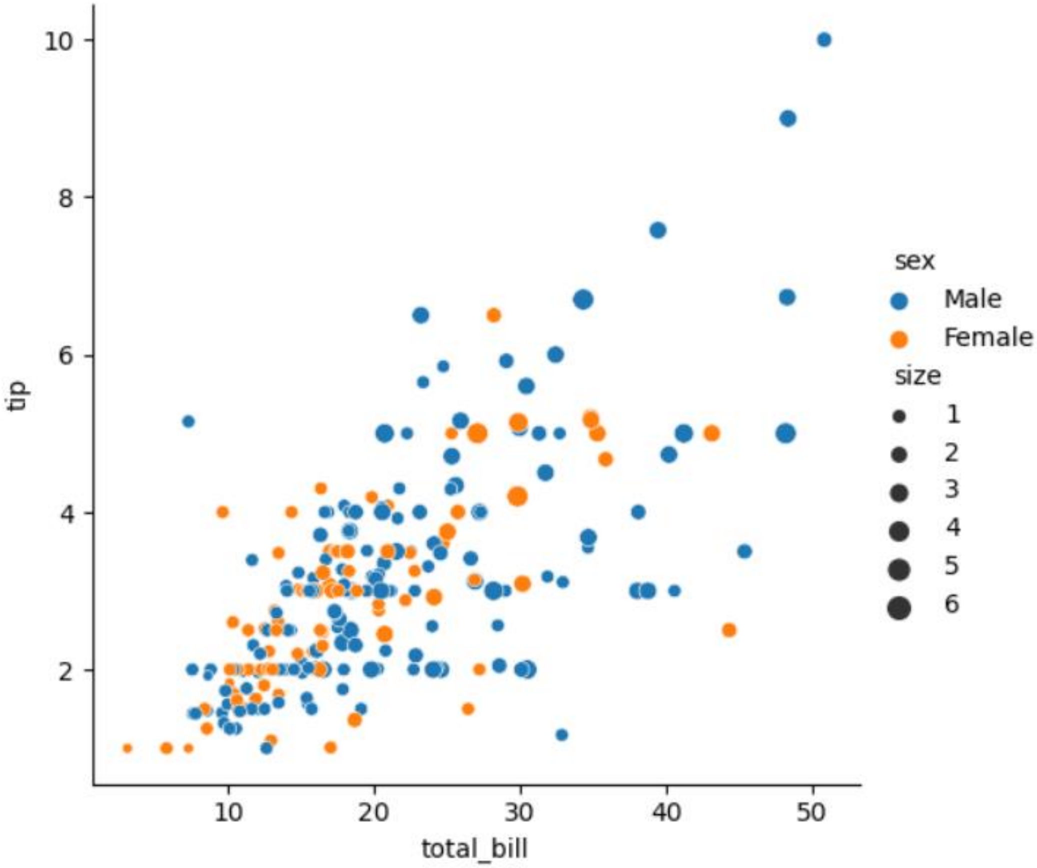
```
ax = sns.scatterplot(x="total_bill", y="tip", data=tips)
```



[66]:

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

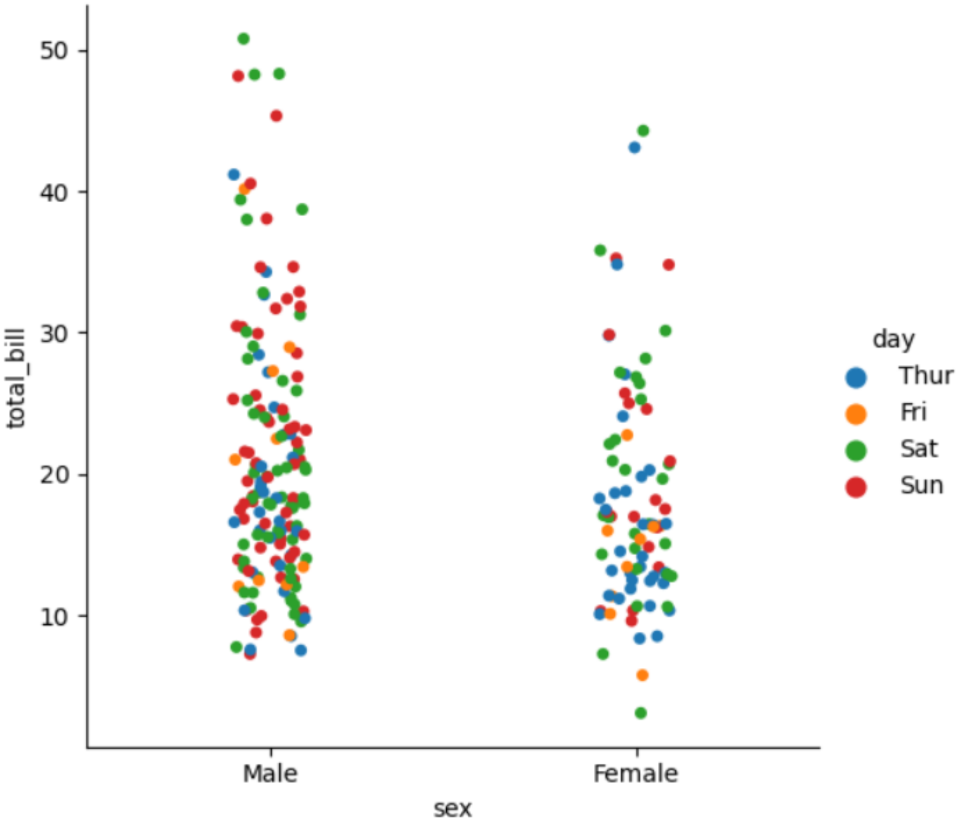
```
[66]: <seaborn.axisgrid.FacetGrid at 0x1f5a7faf760>
```



```
[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")
```

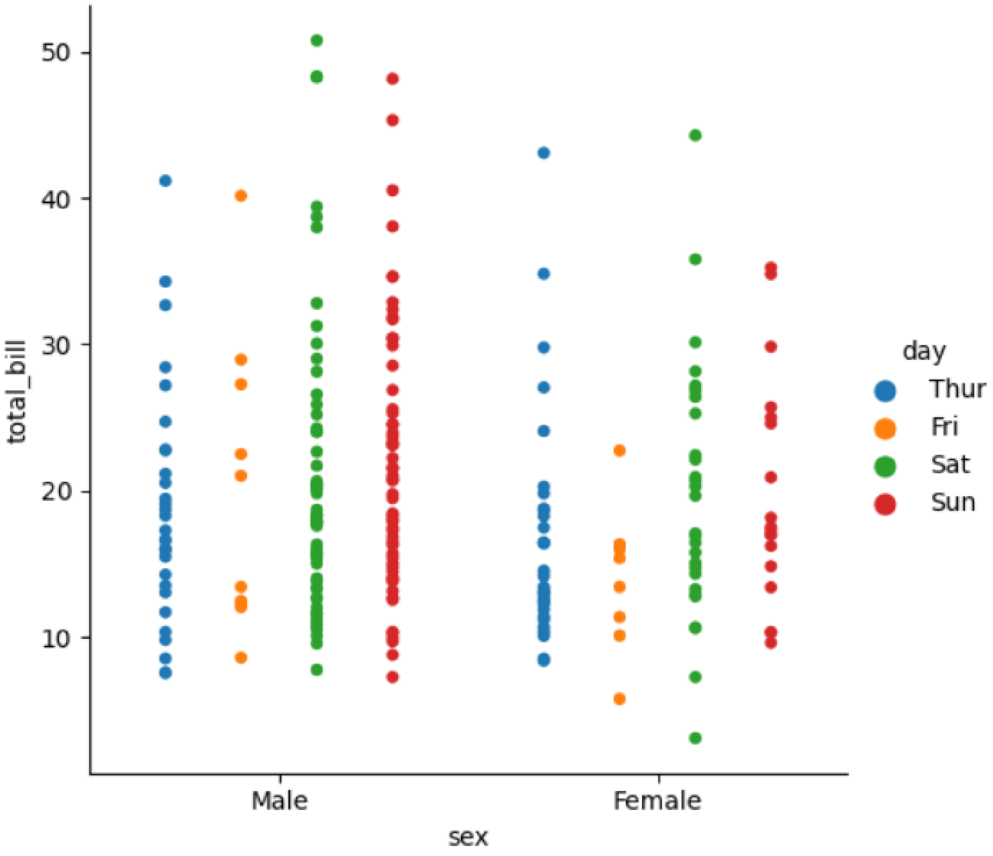
```
[68]: <seaborn.axisgrid.FacetGrid at 0x1f5a7faf340>
```



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

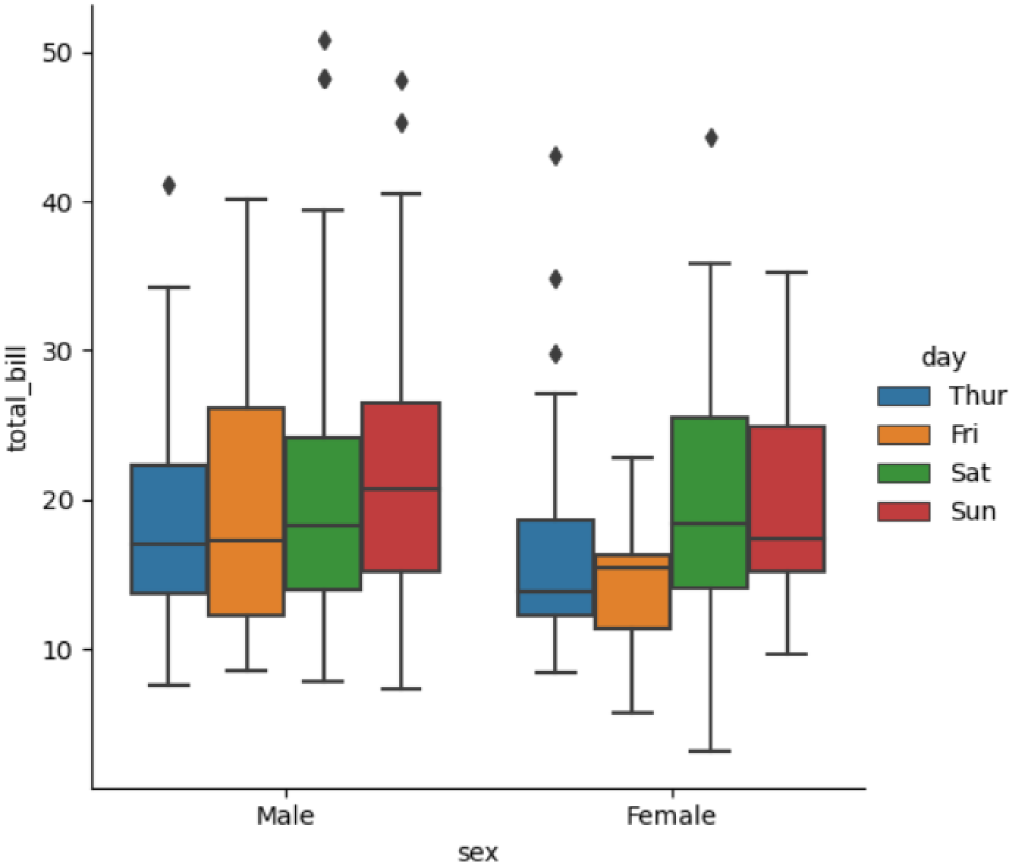
```
[70]: sns.catplot(x="sex", y="total_bill", hue="day", data=tips, kind="strip",
                  jitter=False, dodge=True)
```

```
[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
```

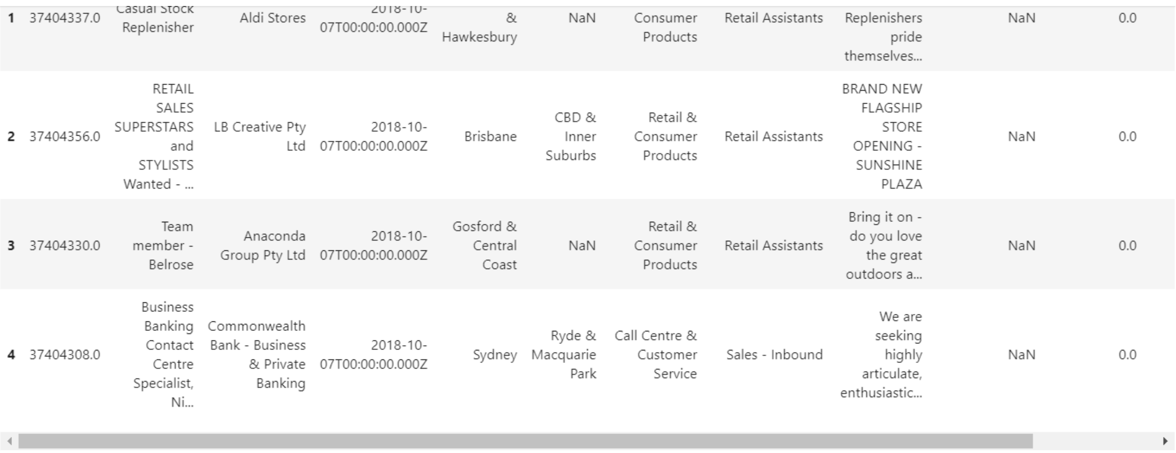
```
[ ]: import matplotlib.pyplot as plt
```

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

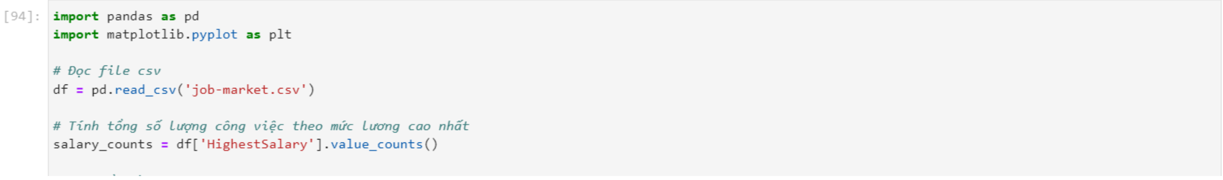
```
[86]: df.head()
```

[86]:

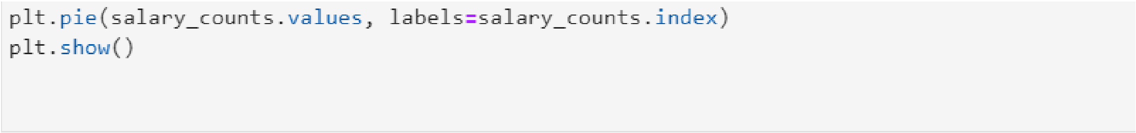
	Id	Title	Company	Date	Location	Area	Classification	SubClassification	Requirement	FullDescription	LowestSalary	Hig
0	37404348.0	Casual Stock Replenisher	Aldi Stores	2018-10-07T00:00:00.000Z	Sydney	North West & Hills District	Retail & Consumer Products	Retail Assistants	Our Casual Stock Replenishers pride 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			



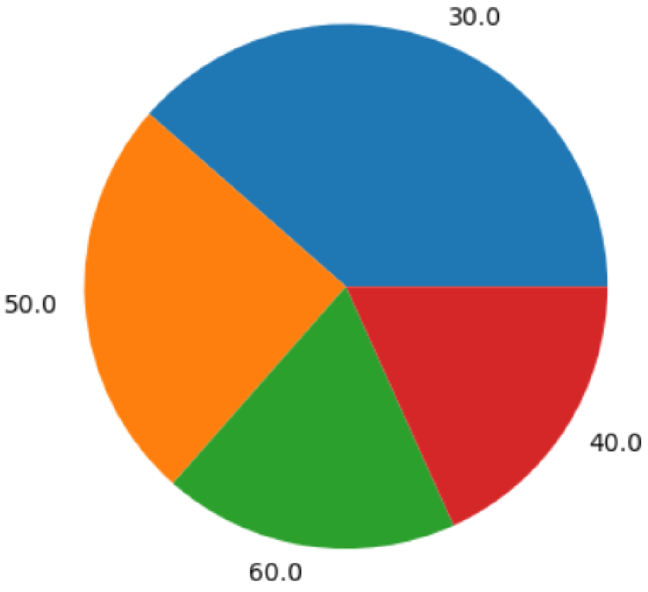
1	37404337.0	Casual Stock Replenisher	Aldi Stores	2018-10-07T00:00:00.000Z	& Hawkesbury	NaN	Consumer Products	Retail Assistants	Replenishers pride themselves...	NaN	0.0
2	37404356.0	RETAIL SALES SUPERSTARS and STYLISTS Wanted - ...	LB Creative Pty Ltd	2018-10-07T00:00:00.000Z	Brisbane	CBD & Inner Suburbs	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

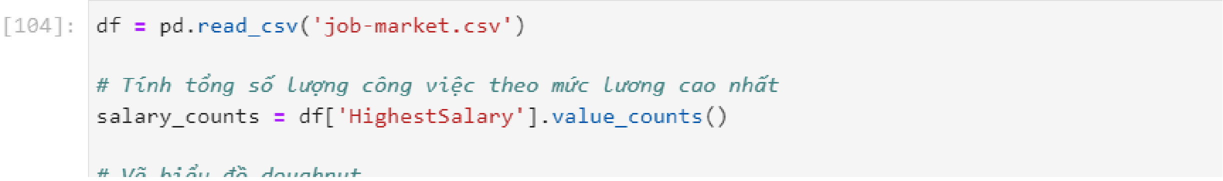


```
[94]: import pandas as pd
      import matplotlib.pyplot as plt

      # 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()
```

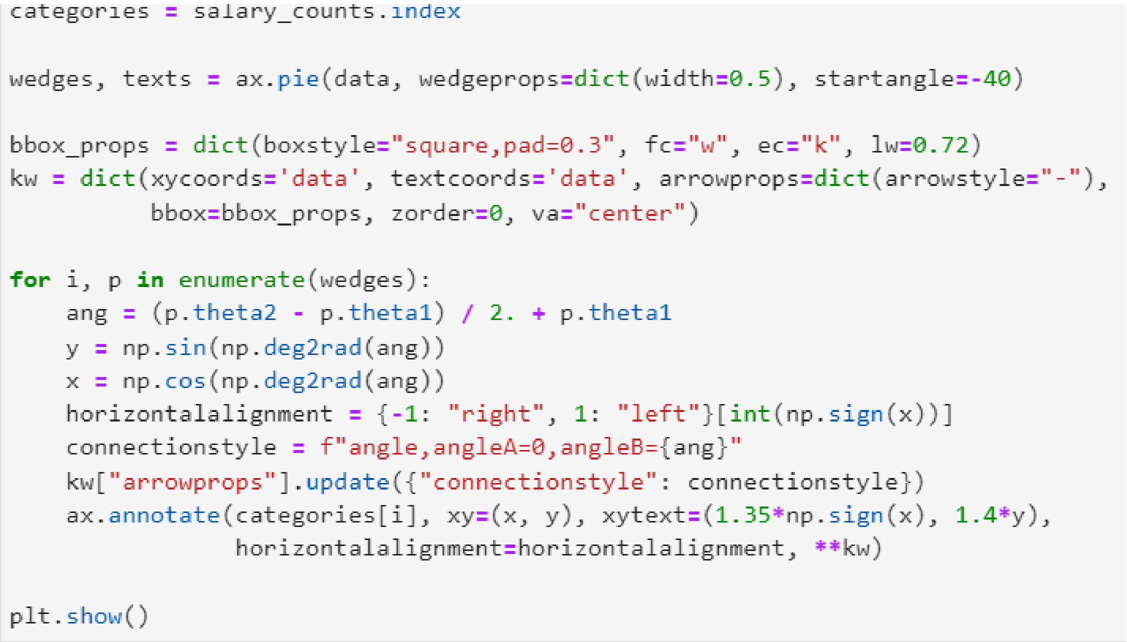


```
plt.pie(salary_counts.values, labels=salary_counts.index)
plt.show()
```

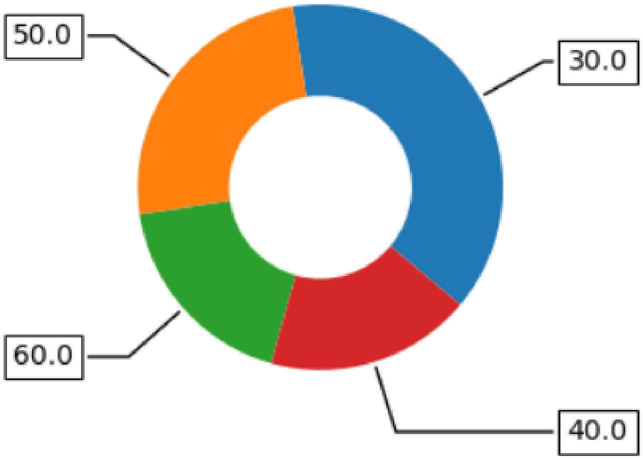




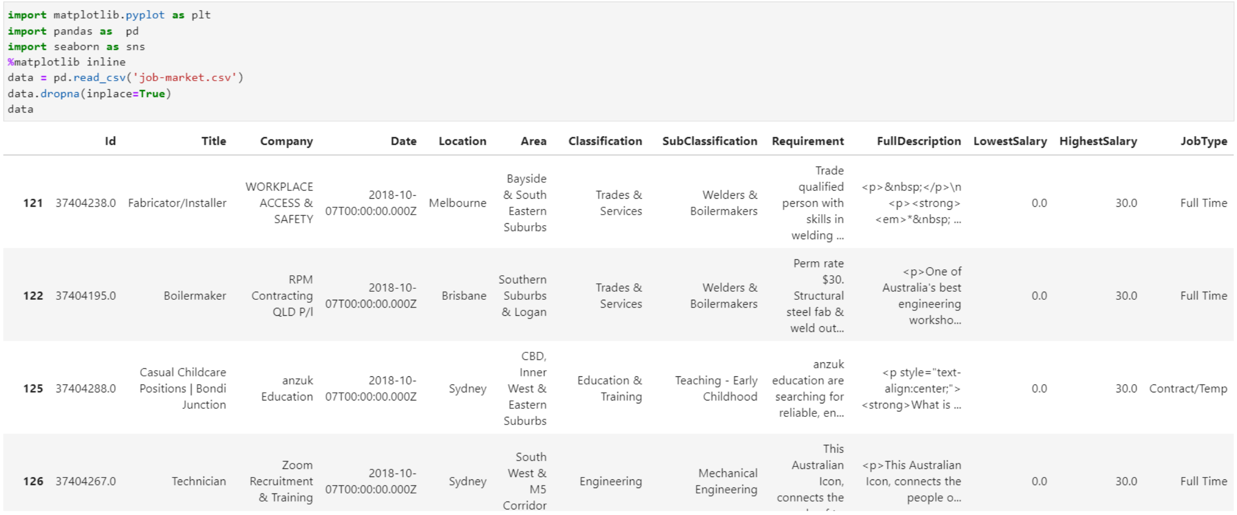
```
[104]: df = pd.read_csv('job-market.csv')

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

       # Vẽ biểu đồ doughnut
```



```
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()
```

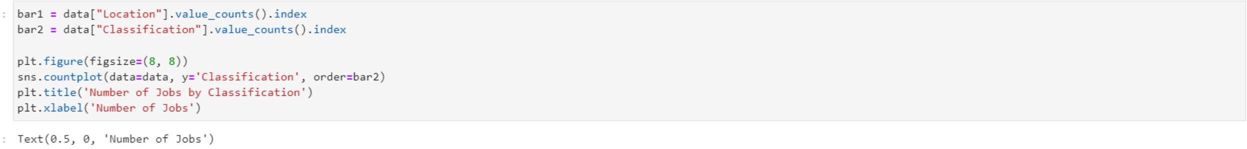


Exercise:



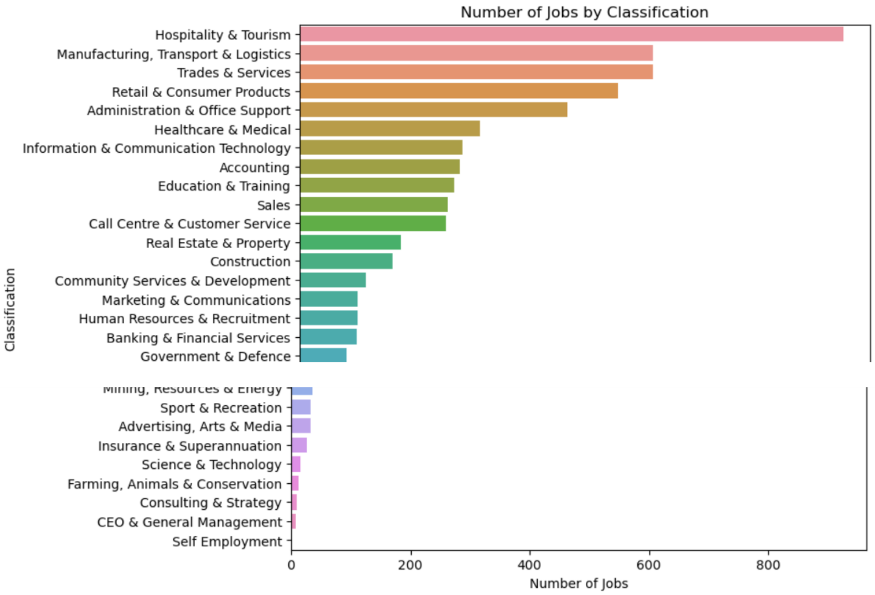
```
import matplotlib.pyplot as plt
import pandas as  pd
import seaborn as sns
%matplotlib inline
data = pd.read_csv('job-market.csv')
data.dropna(inplace=True)
data
```

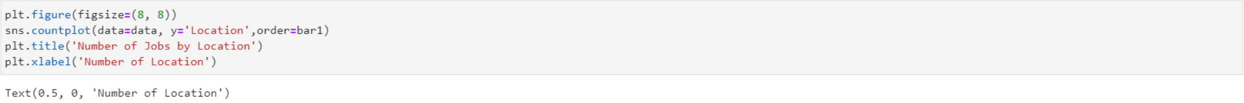
	Id	Title	Company	Date	Location	Area	Classification	SubClassification	Requirement	FullDescription	LowestSalary	HighestSalary	JobType
121	37404238.0	Fabricator/Installer	WORKPLACE ACCESS & SAFETY	2018-10-07T00:00:00.000Z	Melbourne	Bayside & South Eastern Suburbs	Trades & Services	Welders & Boilermakers	Trade qualified person with skills in welding ...	<p>&nbsp;</p>\n<p><strong><em>* ...	0.0	30.0	Full Time
122	37404195.0	Boilermaker	RPM Contracting QLD P/l	2018-10-07T00:00:00.000Z	Brisbane	Southern Suburbs & Logan	Trades & Services	Welders & Boilermakers	Perm rate $30. Structural steel fab & weld out...	<p>One of Australia's best engineering worksho...	0.0	30.0	Full Time
125	37404288.0	Casual Childcare Positions | Bondi Junction	anzuk Education	2018-10-07T00:00:00.000Z	Sydney	CBD, Inner West & Eastern Suburbs	Education & Training	Teaching - Early Childhood	anzuk education are searching for reliable, en...	<p style="text-align:center;"><strong>What is ...	0.0	30.0	Contract/Temp
126	37404267.0	Technician	Zoom Recruitment & Training	2018-10-07T00:00:00.000Z	Sydney	South West & M5 Corridor	Engineering	Mechanical Engineering	This Australian Icon, connects the	<p>This Australian Icon, connects the people o...	0.0	30.0	Full Time



```
bar1 = data["Location"].value_counts().index
bar2 = data["Classification"].value_counts().index

plt.figure(figsize=(8, 8))
sns.countplot(data=data, y='Classification', order=bar2)
plt.title('Number of Jobs by Classification')
plt.xlabel('Number of Jobs')
```

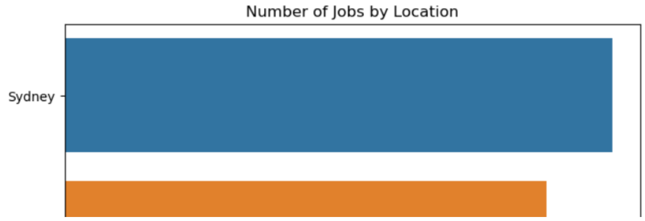
```
Text(0.5, 0, 'Number of Jobs')
```

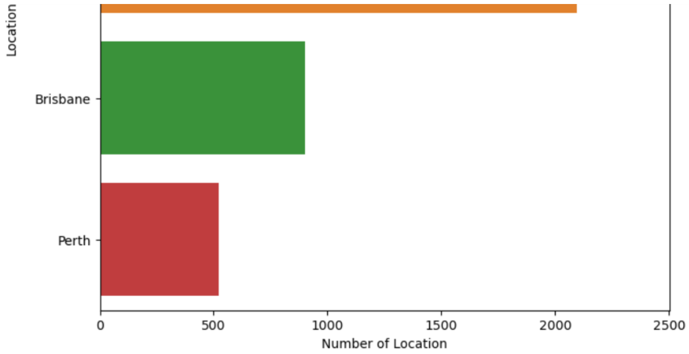


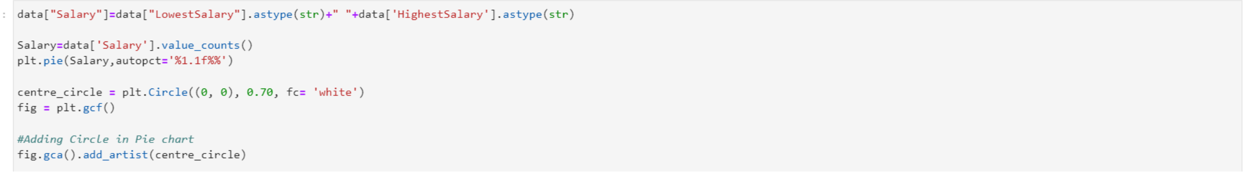


```
plt.figure(figsize=(8, 8))
sns.countplot(data=data, y='Location',order=bar1)
plt.title('Number of Jobs by Location')
plt.xlabel('Number of Location')
```

```
Text(0.5, 0, 'Number of Location')
```

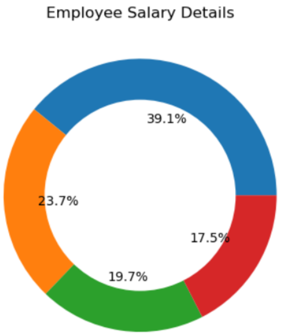






```
data["Salary"]=data["LowestSalary"].astype(str)+" "+data['HighestSalary'].astype(str)

Salary=data['Salary'].value_counts()
plt.pie(Salary,autopct='%1.1f%%')

centre_circle = plt.Circle((0, 0), 0.70, fc= 'white')
fig = plt.gcf()

#Adding Circle in Pie chart
fig.gca().add_artist(centre_circle)
```

```
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()

#Adding Circle in Pie chart
fig.gca().add_artist(centre_circle)

#Adding Title of chart
plt.title('Employee Salary Details')

#Displaying Chart
plt.show()
```



```
Salary=data['Salary'].value_counts()
Salary
```

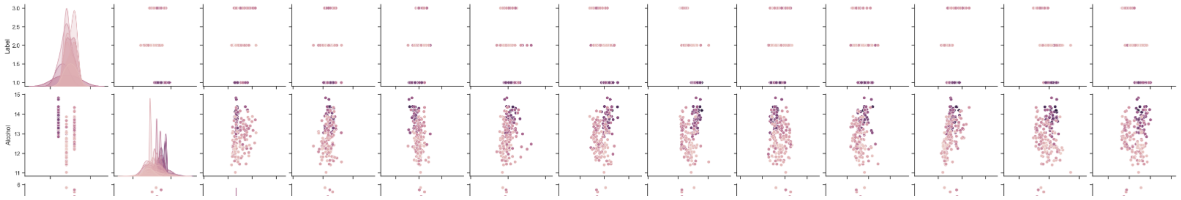
```
0.0 30.0     2309
40.0 50.0    1397
50.0 60.0    1161
30.0 40.0    1031
Name: Salary, dtype: int64
```

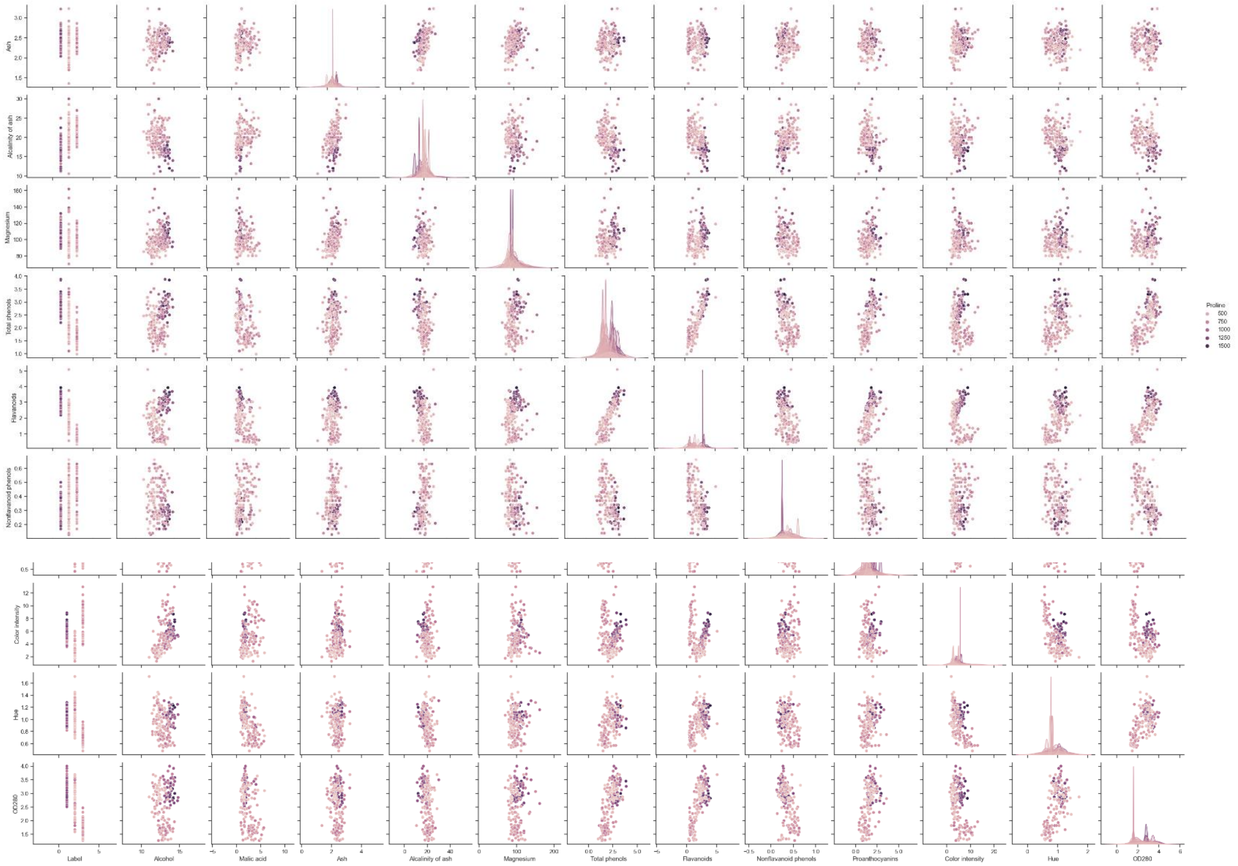
```
import pandas as pd

df = pd.read_csv('wine.data.csv')
print(df)
```

```
     Label  Alcohol  Malic acid   Ash  Alcalinity of ash  Magnesium  \
0        1    14.23        1.71  2.43               15.6        127   
1        1    13.20        1.78  2.14               11.2        100   
2        1    13.16        2.36  2.67               18.6        101   
3        1    14.37        1.95  2.50               16.8        113   
4        1    13.24        2.59  2.87               21.0        118   
..     ...      ...         ...   ...                ...        ...   
173      3    13.71        5.65  2.45               20.5         95   
174      3    13.40        3.91  2.48               23.0        102   
175      3    13.27        4.28  2.26               20.0        120   
176      3    13.17        2.59  2.37               20.0        120   
177      3    14.13        4.10  2.74               24.5         96   

     Total phenols  Flavanoids  Nonflavanoid phenols  Proanthocyanins  \
0             2.80        3.06                  0.28             2.29   
1             2.65        2.76                  0.26             1.28   
2             2.80        3.24                  0.30             2.81   
3             3.85        3.49                  0.24             2.18   
4             2.80        2.69                  0.39             1.82   
..             ...         ...                   ...              ...   
173           1.68        0.61                  0.52             1.06   

      Color intensity   Hue  OD280  Proline  
 0               5.64  1.04   3.92     1065  
 1               4.38  1.05   3.40     1050  
 2               5.68  1.03   3.17     1185  
 3               7.80  0.86   3.45     1480  
 4               4.32  1.04   2.93      735  
 ..               ...   ...    ...      ...  
 173             7.70  0.64   1.74      740  
 174             7.30  0.70   1.56      750  
 175            10.20  0.59   1.56      835  
 176             9.30  0.60   1.62      840  
 177             9.20  0.61   1.60      560  

 [178 rows x 14 columns]
```

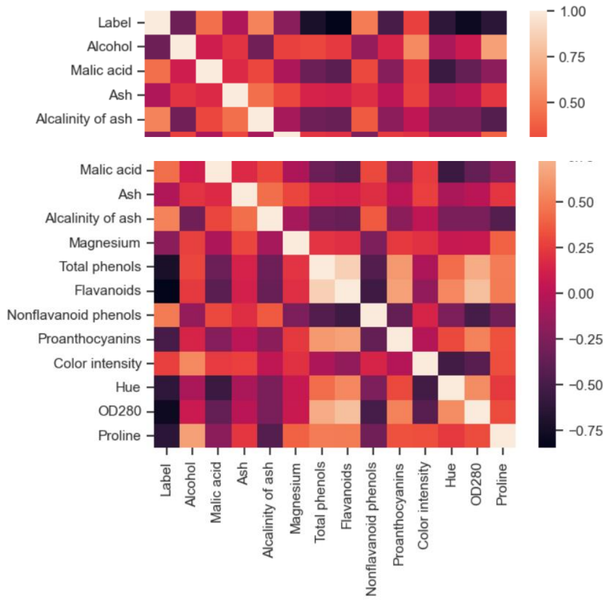
```
]: df = df.drop('Label', axis=1)
   titanic_dataset = pd.read_csv('wine.data.csv')
   sns.set_theme(style="ticks")
   sns.pairplot(titanic_dataset, hue='Proline')
```

```
]: <seaborn.axisgrid.PairGrid at 0x26ca7bc2520>
```





```
import matplotlib.pyplot as mp
import pandas as pd
import seaborn as sb
dataplot=sb.heatmap(titanic_dataset.corr())
```


Label
Alcohol
Malic acid
Ash
Alcalinity of ash
Malic acid
Ash
Alcalinity of ash
Magnesium
Total phenols
Flavanoids
Nonflavanoid phenols
Proanthocyanins
Color intensity
Hue
OD280
Proline
Label
Alcohol
Malic acid
Ash
Alcalinity of ash
Magnesium
Total phenols
Flavanoids
Nonflavanoid phenols
Proanthocyanins
Color intensity
Hue
OD280
Proline