Le Ngoc Thai Phuong - 18521272

I. Matplotlib

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

In [2]:
```
1 %matplotlib inline
```

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

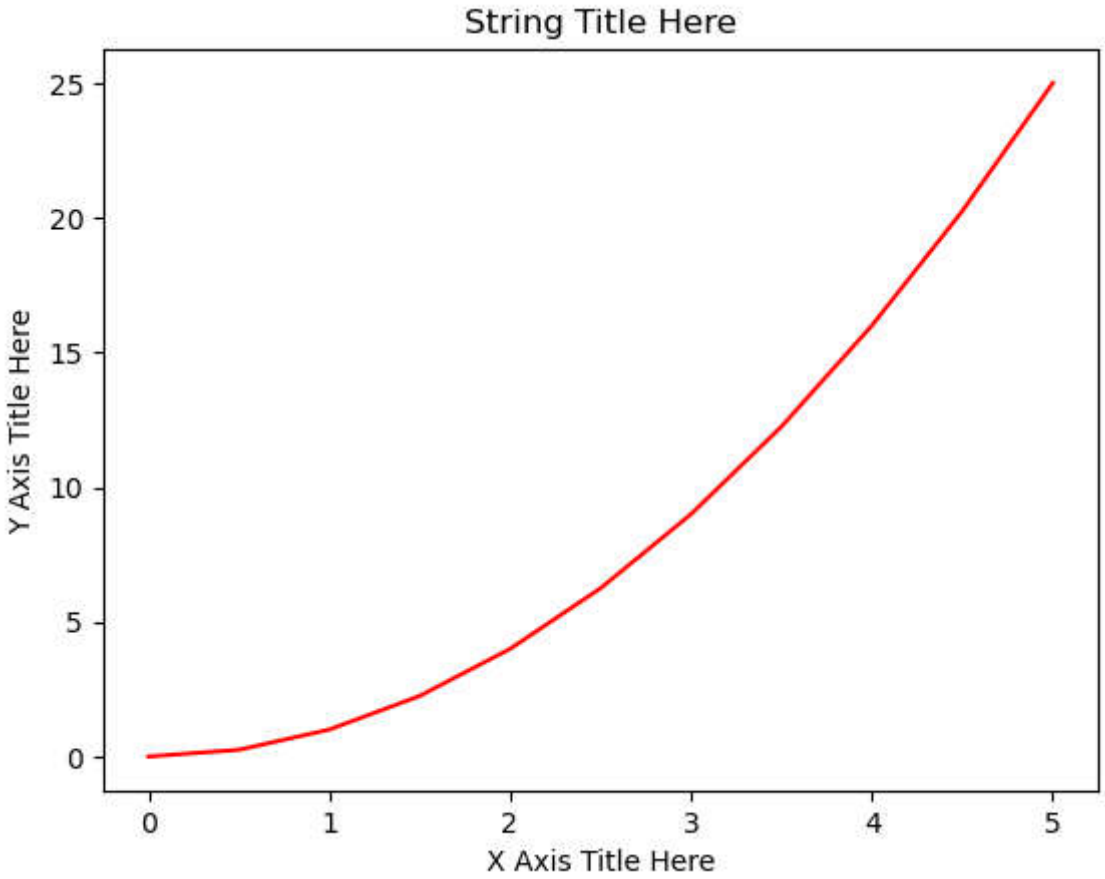
In [4]:
```
1 x
```

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

In [5]:
```
1 y
```

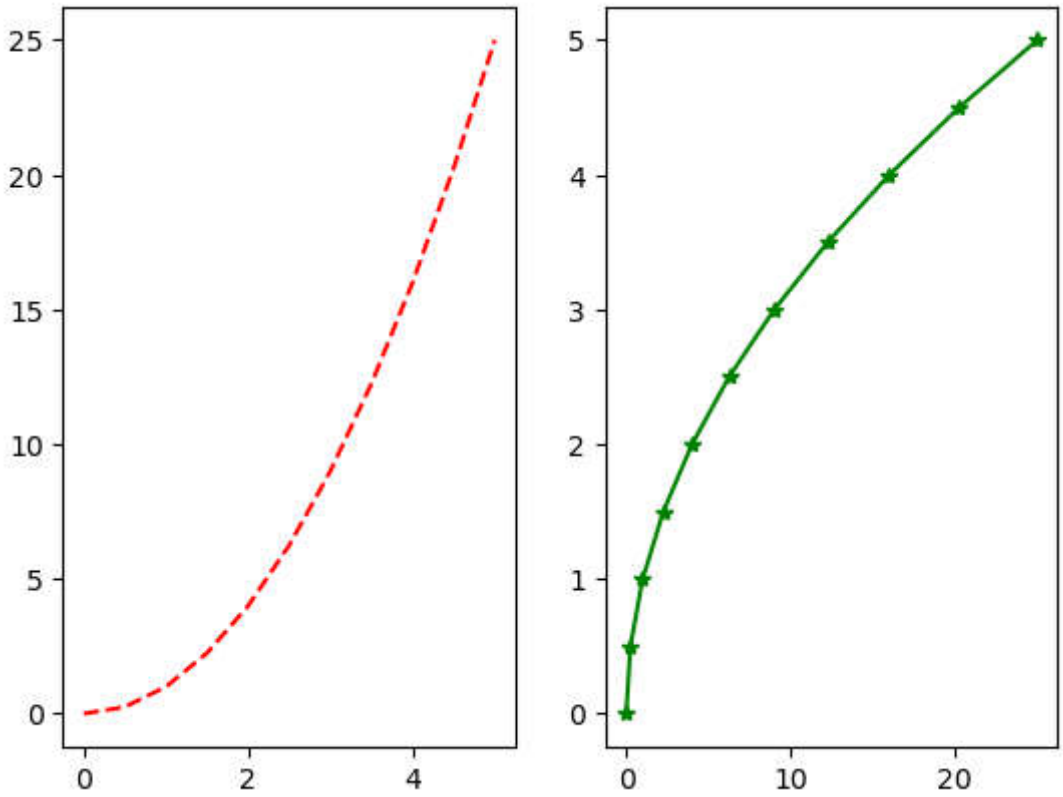
Out[5]:
```
array([ 0.  ,  0.25,  1.  ,  2.25,  4.  ,  6.25,  9.  , 12.25, 16.  ,
       20.25, 25.  ])
```

In [6]:
```
1 plt.plot(x, y, 'r') # 'r' is the color red
2 plt.xlabel('X Axis Title Here')
3 plt.ylabel('Y Axis Title Here')
4 plt.title('String Title Here')
5 plt.show()
```



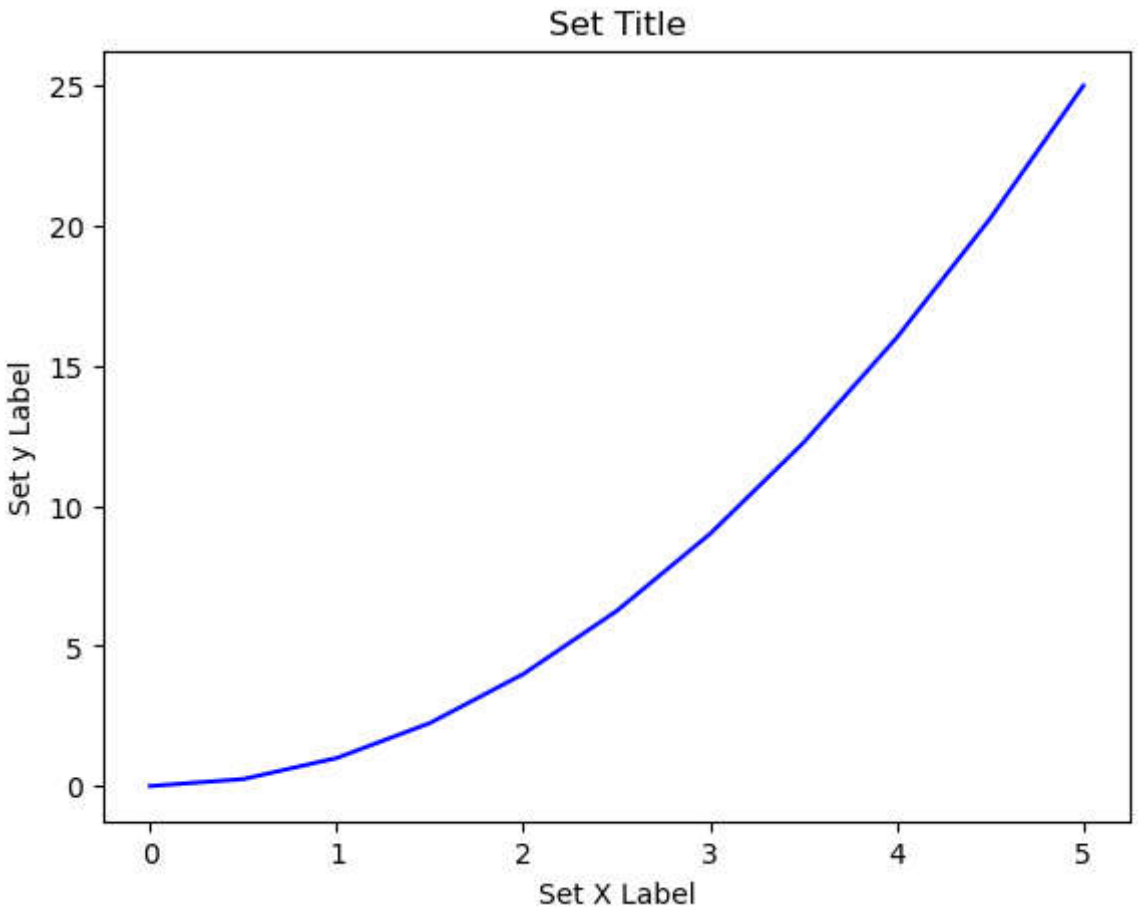
In [7]:

```
1
2 plt.subplot(1,2,1)
3 plt.plot(x, y, 'r--')
4 plt.subplot(1,2,2)
5 plt.plot(y, x, 'g*-');
```



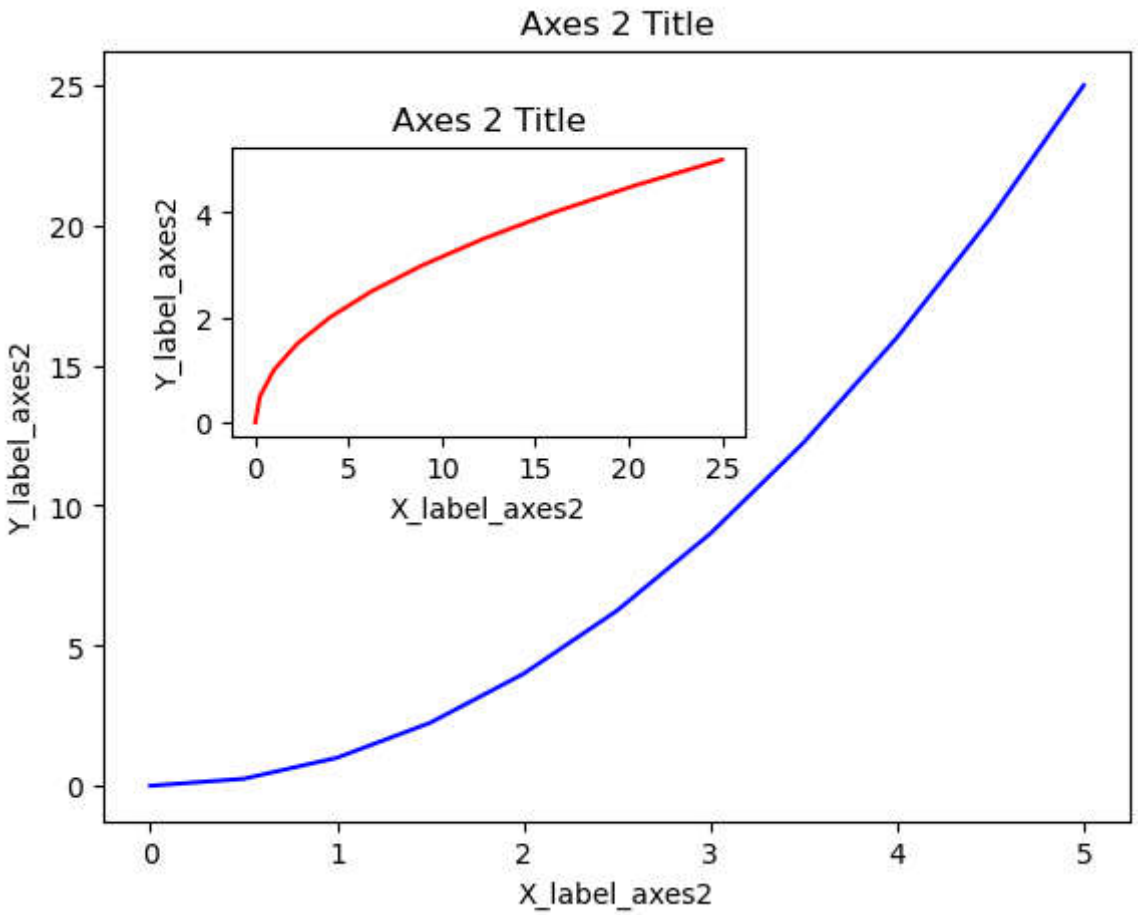
In [8]:
```
# Create Figure
fig = plt.figure()

# 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')
axes.set_ylabel('Set y Label')
axes.set_title('Set Title')
```

Out[8]: Text(0.5, 1.0, 'Set Title')

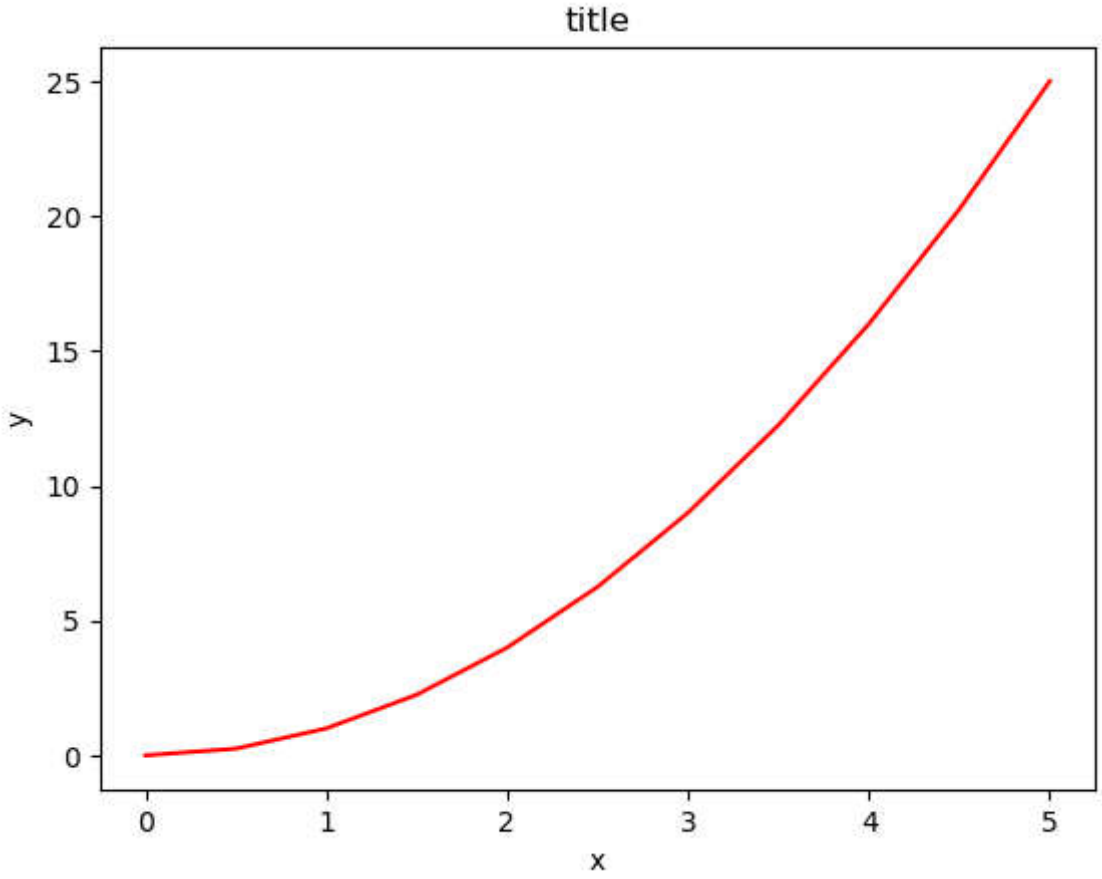


In [9]:

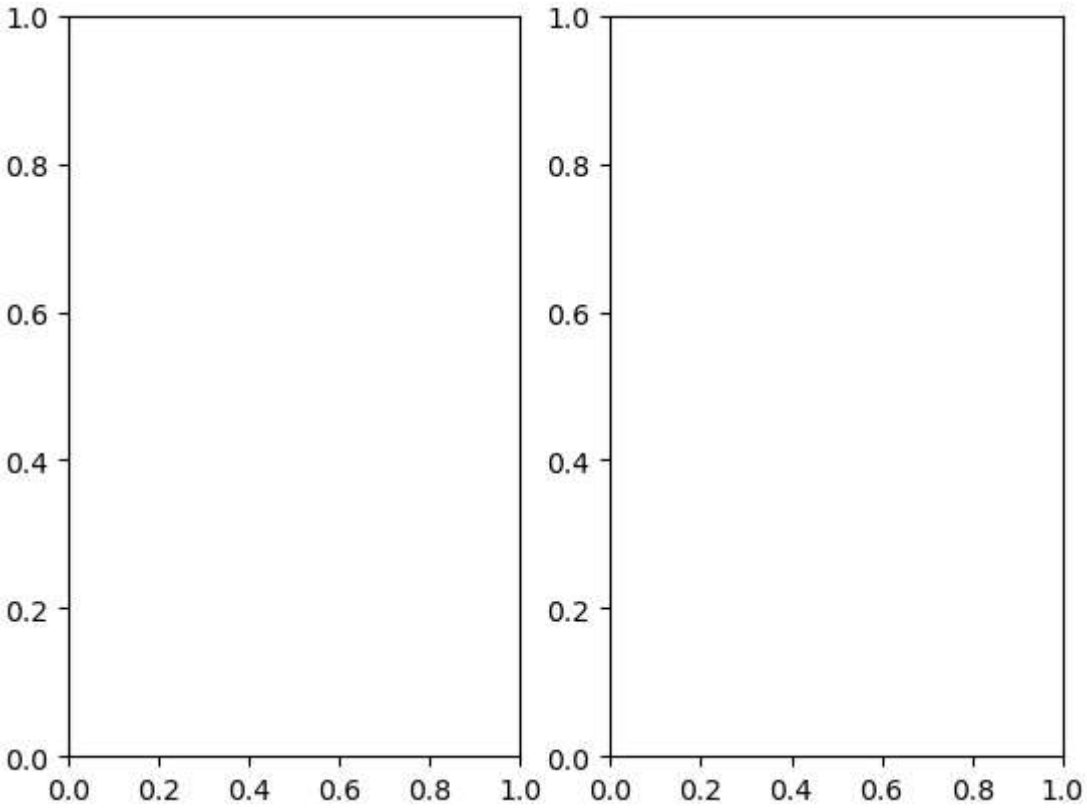
```
fig = plt.figure()

axes1 = fig.add_axes([0.1, 0.1, 0.8, 0.8])
axes2 = fig.add_axes([0.2, 0.5, 0.4, 0.3])


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

#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');
```



In [10]:
```
fig, axes = plt.subplots()

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



In [11]:
```
1 fig, axes = plt.subplots(nrows=1, ncols=2)
```

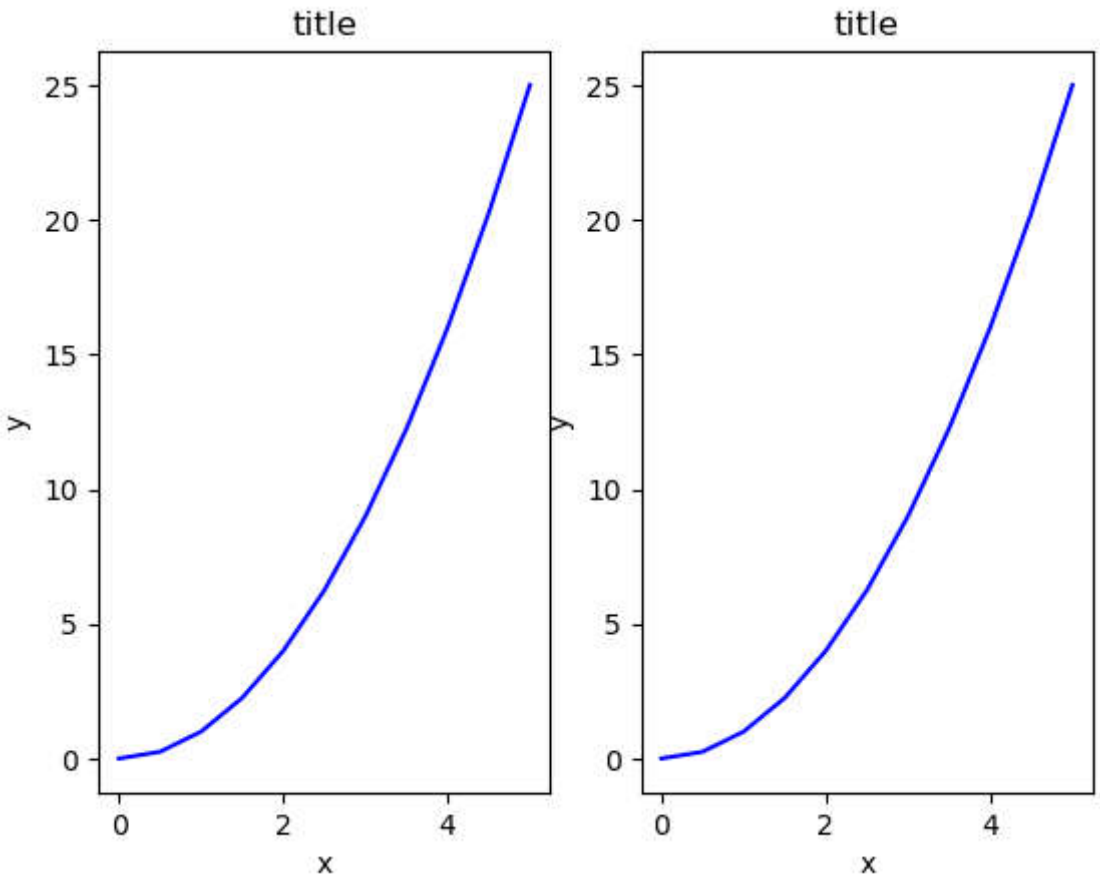


In [12]:
```
1 axes
```

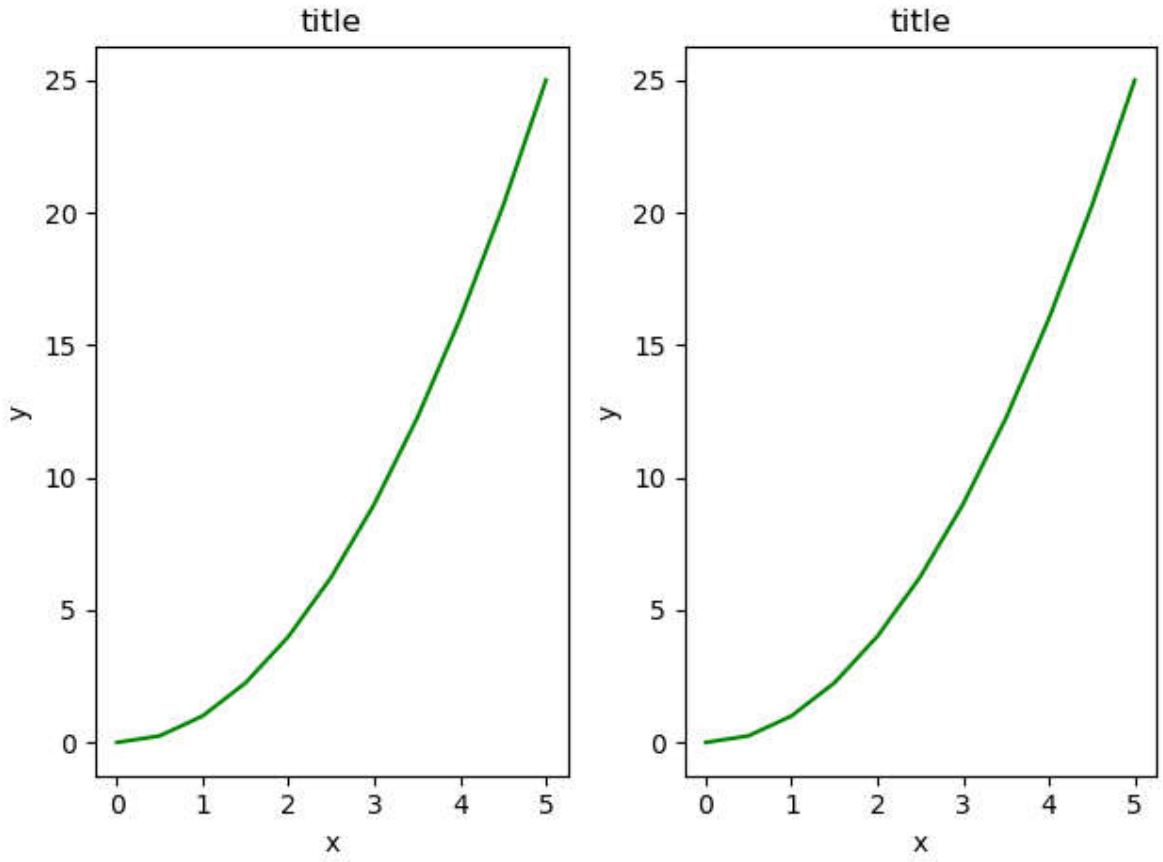
Out[12]:
```
array([<AxesSubplot:>, <AxesSubplot:>], dtype=object)
```

In [13]:
```
for ax in axes:
    ax.plot(x, y, 'b')
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_title('title')

fig
```

Out[13]:



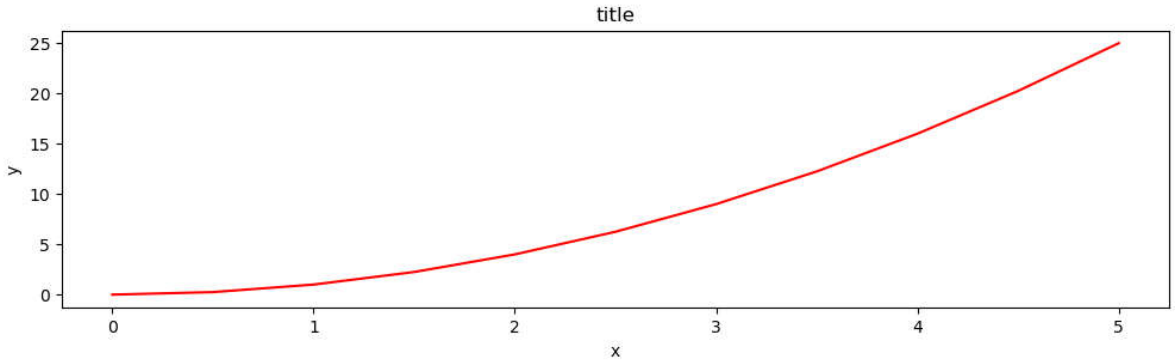
In [14]:
```
fig, axes = plt.subplots(nrows=1, ncols=2)

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()
```



In [15]:
```
fig = plt.figure(figsize=(8,4), dpi =100)
```

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

In [16]:
```
fig, axes = plt.subplots(figsize=(12,3))

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



In [17]:
```
fig.savefig("filename.png")
```

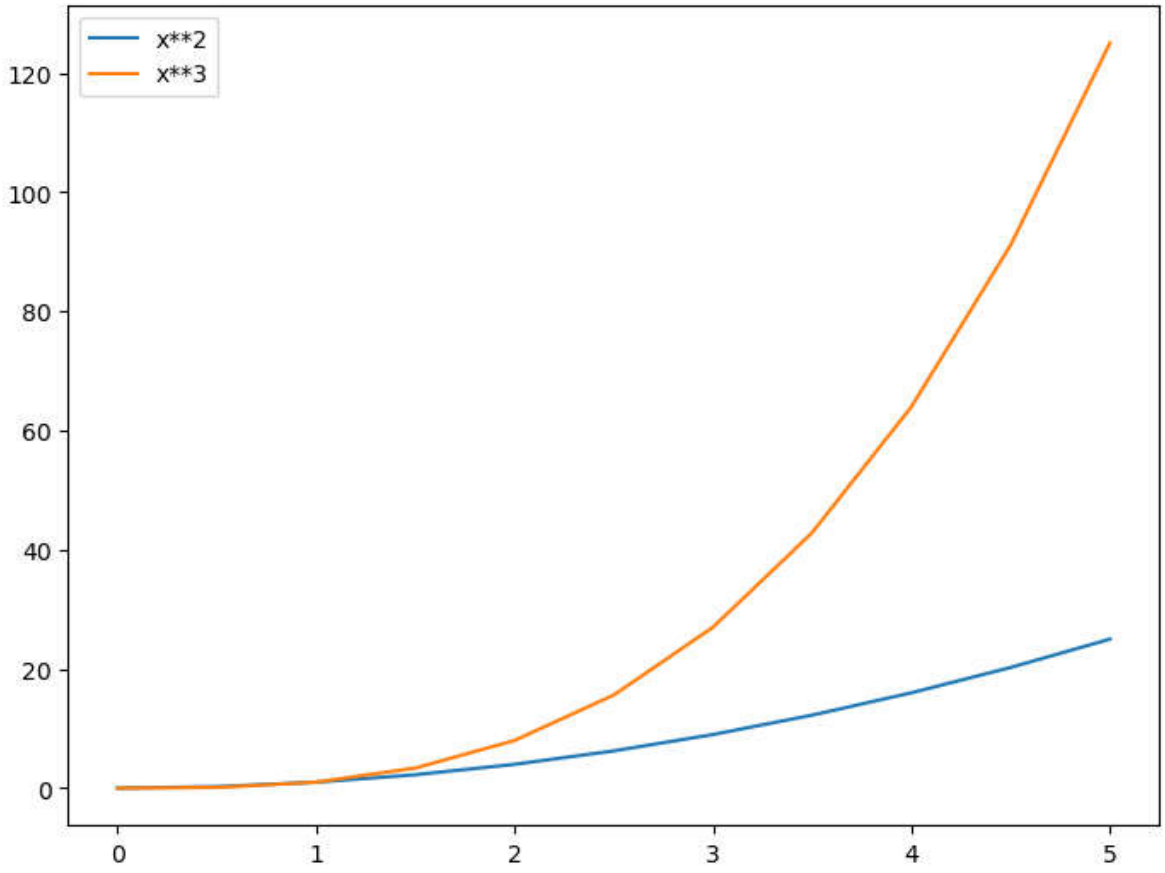
In [18]:
```
fig.savefig("filename.png", dpi=200)
```

In [19]:
```
ax.set_title("title");
```

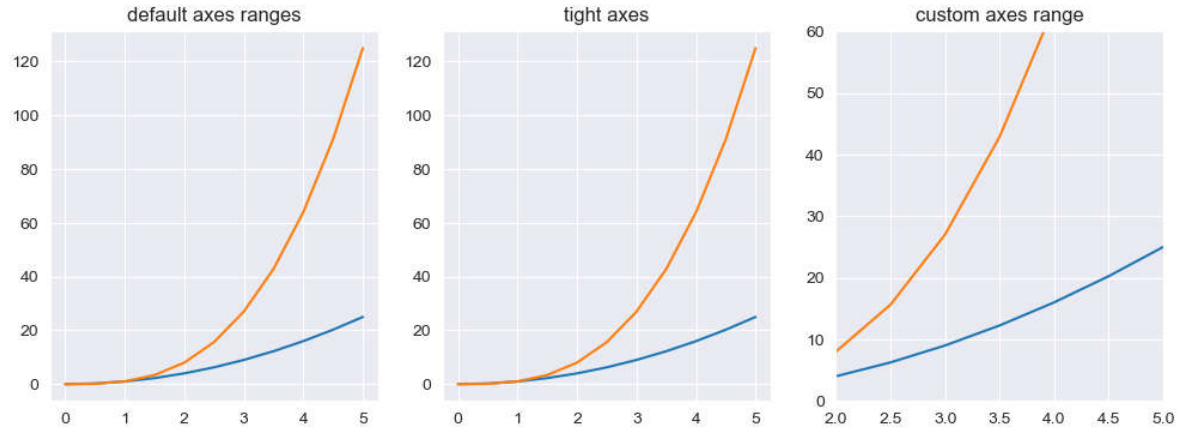
In [20]:
```
ax.set_xlabel("x")
ax.set_ylabel("y");
```

In [21]:
```
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()
```

Out[21]: <matplotlib.legend.Legend at 0x1fde3cf9820>



In [35]:
```
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");
```



II. Seaborn

In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline
sns.get_dataset_names()

```

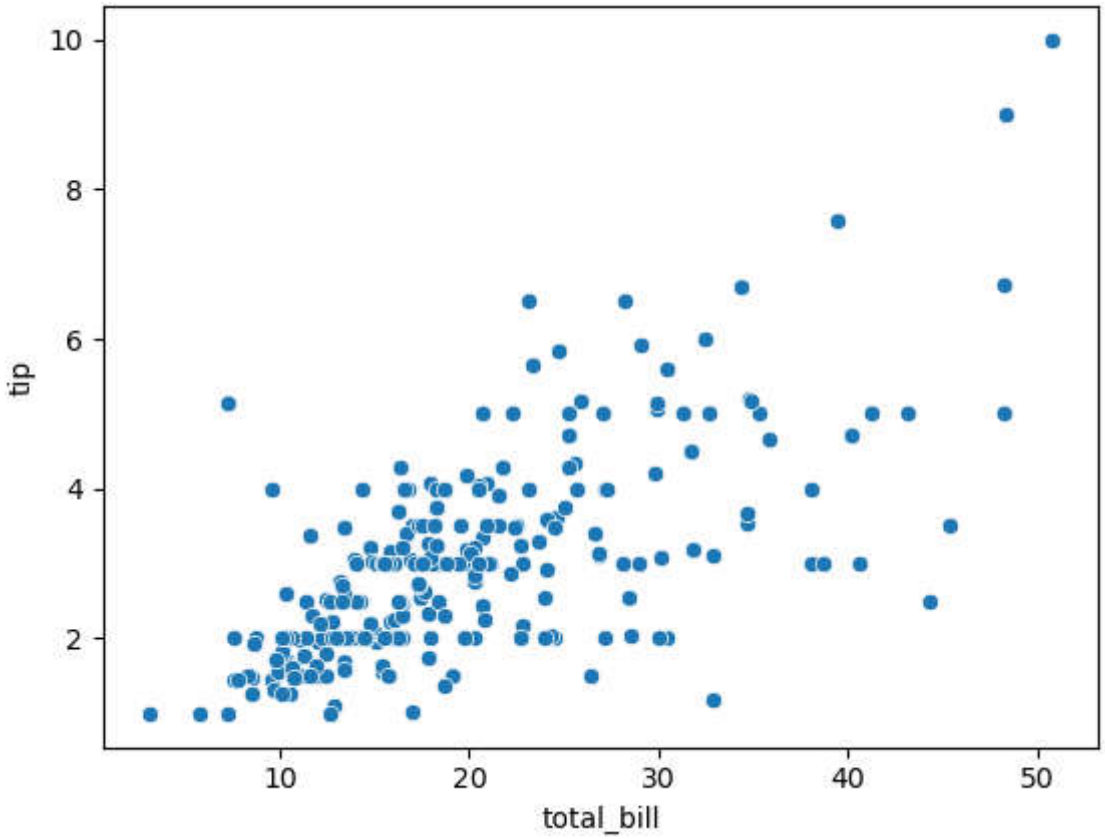
Out[23]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

In [24]:
```
tips = sns.load_dataset("tips")
tips.head()
```

Out[24]:

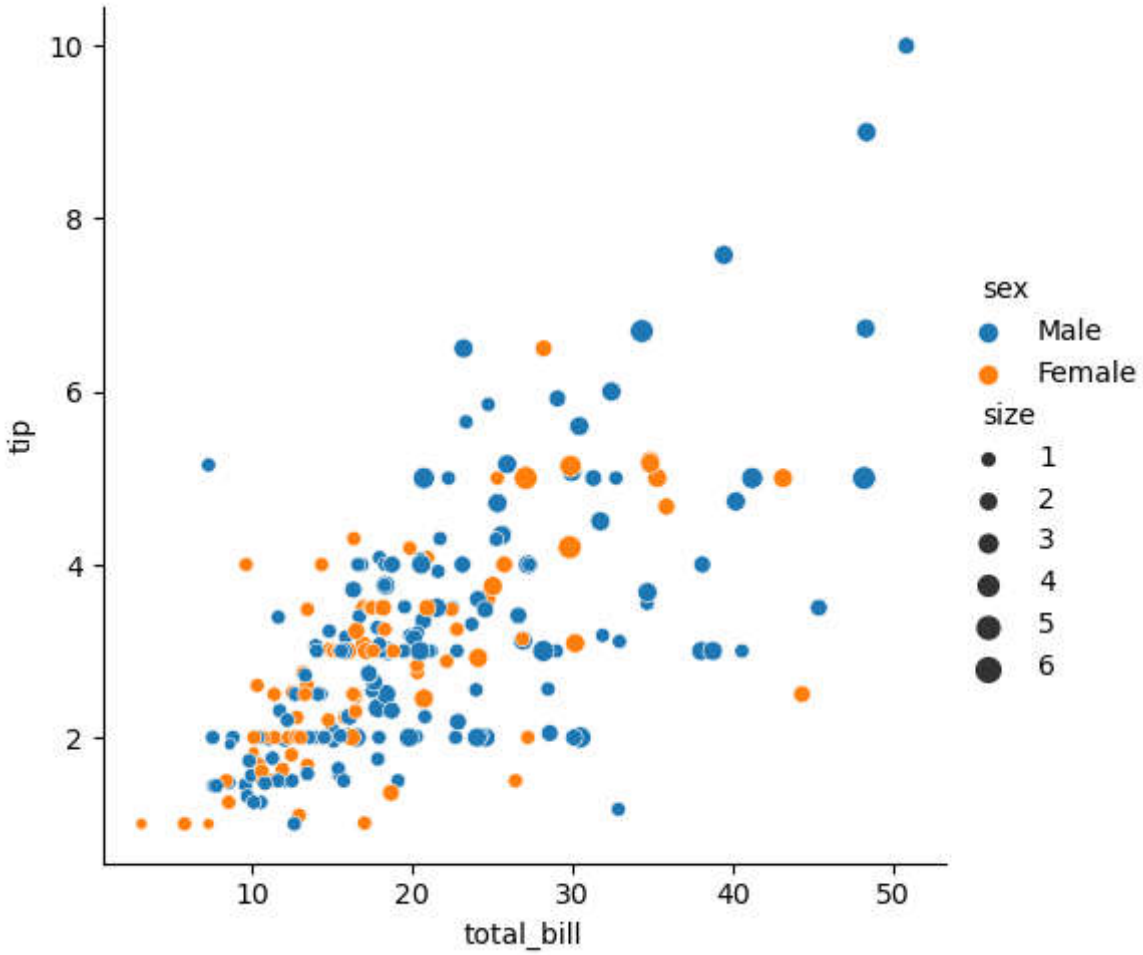
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [25]:
```
1 ax = sns.scatterplot(x="total_bill", y= "tip", data=tips)
```

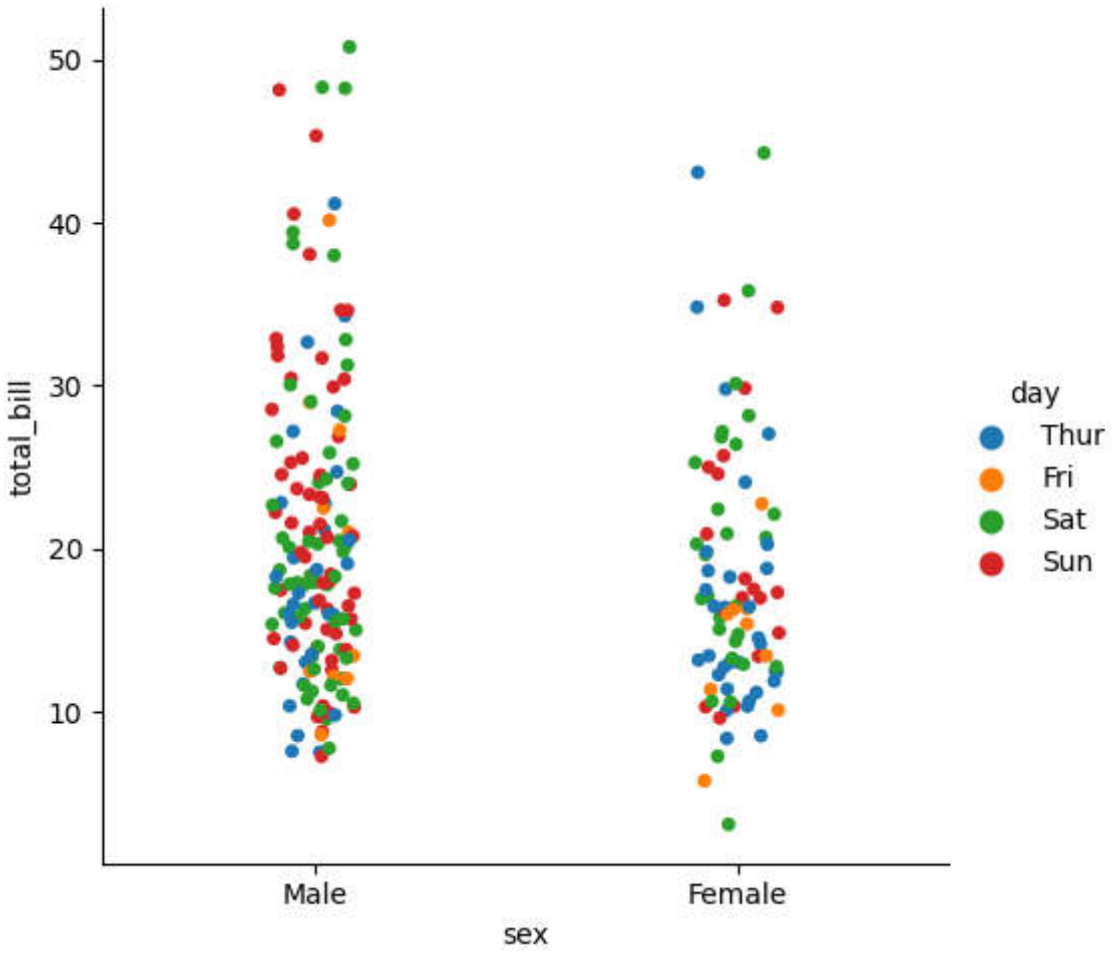


In [26]:
```
sns.relplot(x = "total_bill", y = "tip", data = tips, kind="scatter",
            hue="sex", size="size",
           )
```

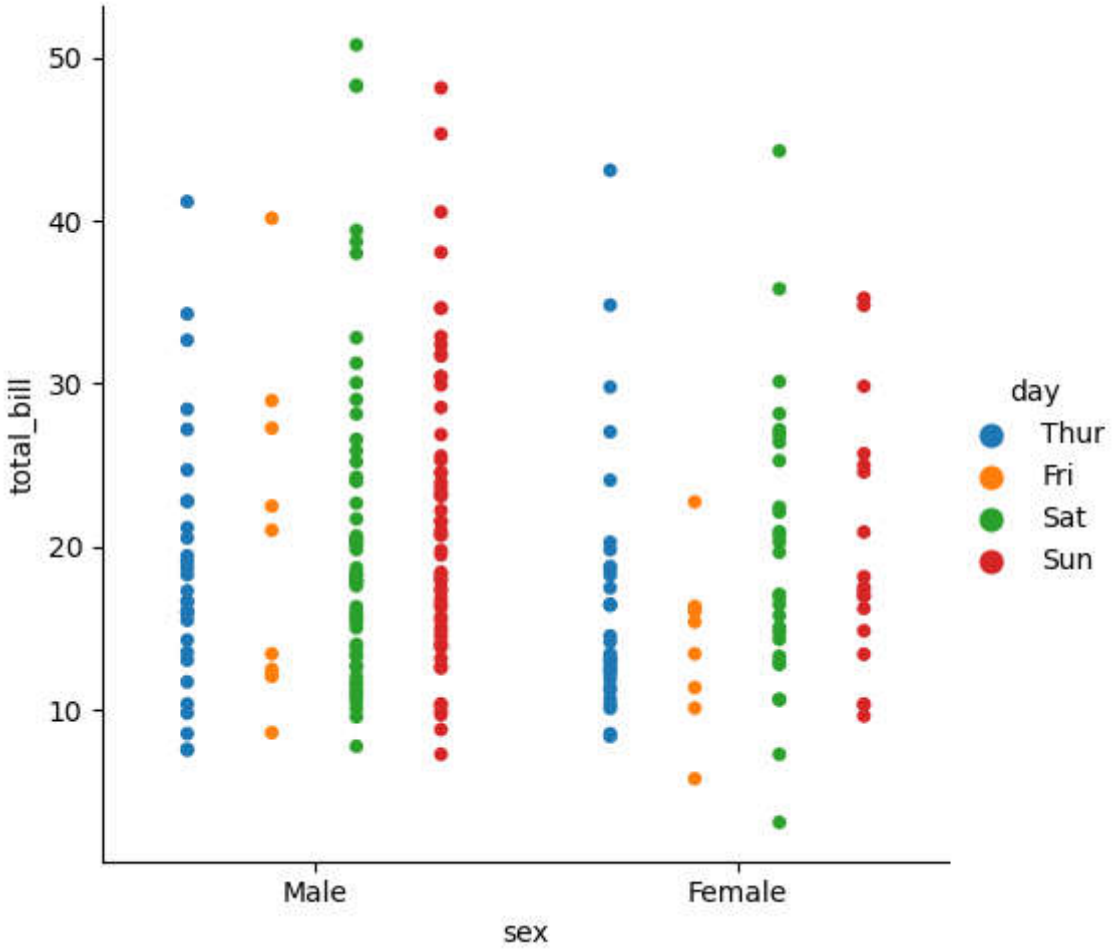
Out[26]: <seaborn.axisgrid.FacetGrid at 0x1fde7215d60>



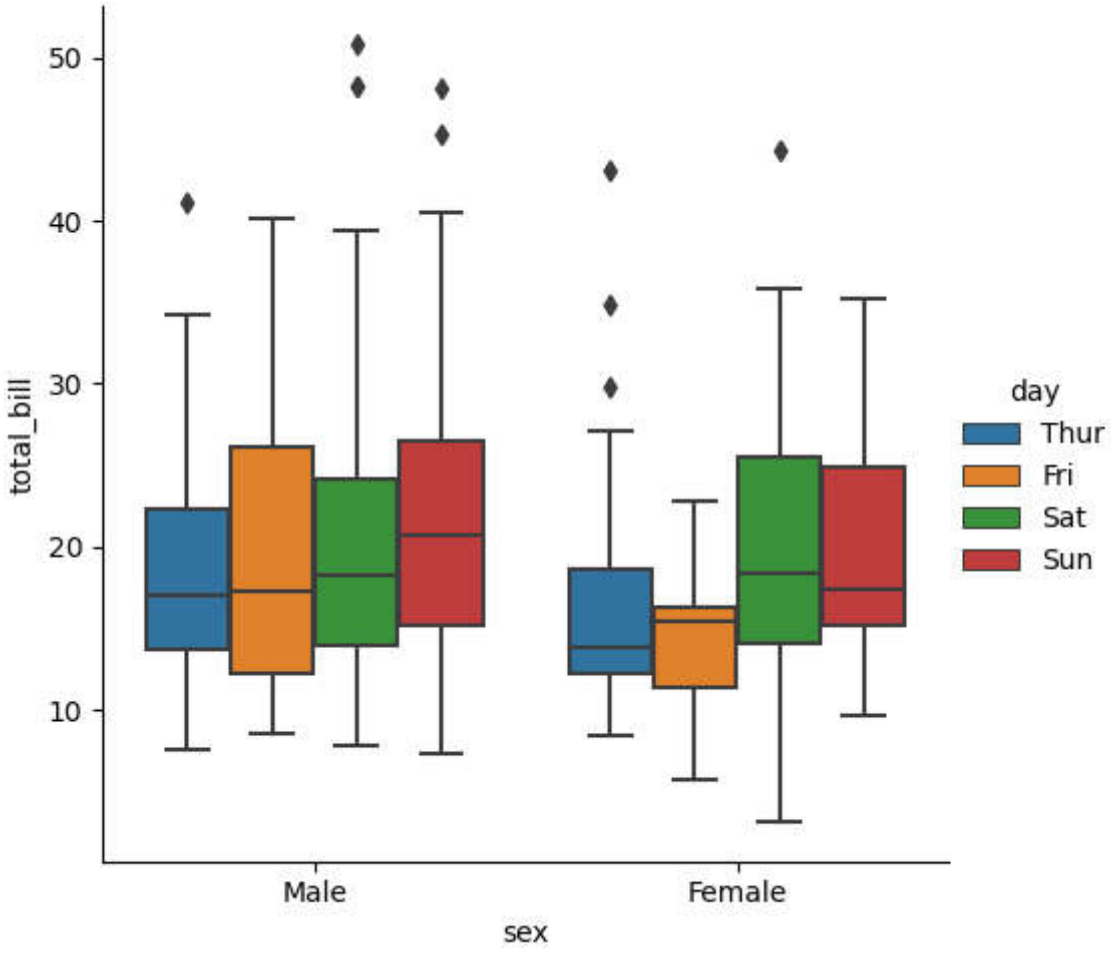
In [27]:
```
1 sns.catplot(x = "sex", y = "total_bill", hue= "day", data=tips, kind="str
```



In [28]:
```
sns.catplot(x = "sex", y = "total_bill", hue= "day", data=tips, kind="str
            jitter=False, dodge=True);
```



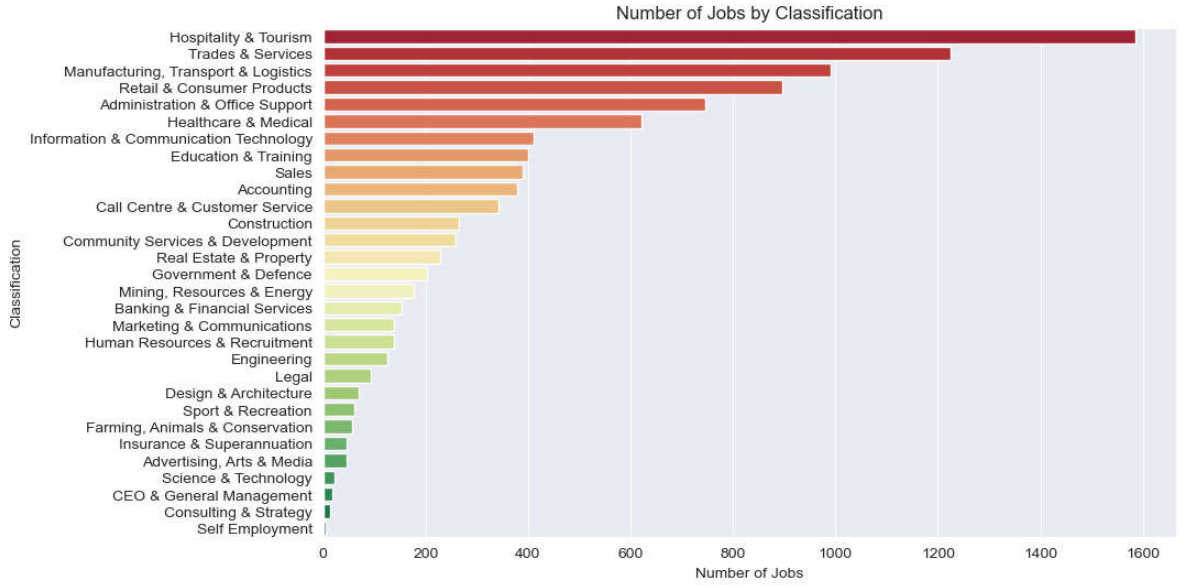
In [29]:
```
1 sns.catplot(x = "sex", y = "total_bill", hue= "day", data=tips, kind="box
```



III. Exercises

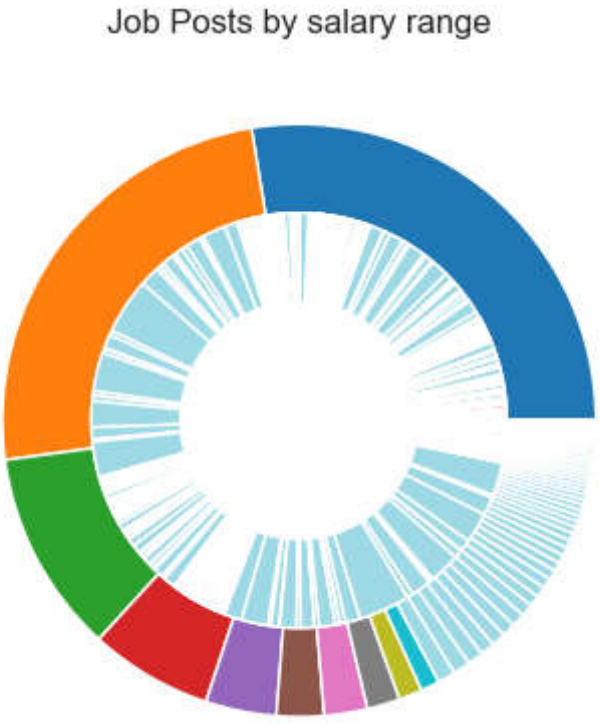
In [36]:

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

df = pd.read_csv("job-market.csv")

# Count the number of jobs by location
jobs_by_location = df.groupby("Classification")["Title"].count()

# Sort the locations by number of jobs in descending order
jobs_by_location = jobs_by_location.sort_values(ascending=False)

# Create a bar chart of the jobs by location
sns.set_style("darkgrid")
plt.figure(figsize=(10, 6))
sns.barplot(x=jobs_by_location.values, y=jobs_by_location.index, palette='
plt.xlabel("Number of Jobs")
plt.ylabel("Classification")
plt.title("Number of Jobs by Classification")
plt.show()

```



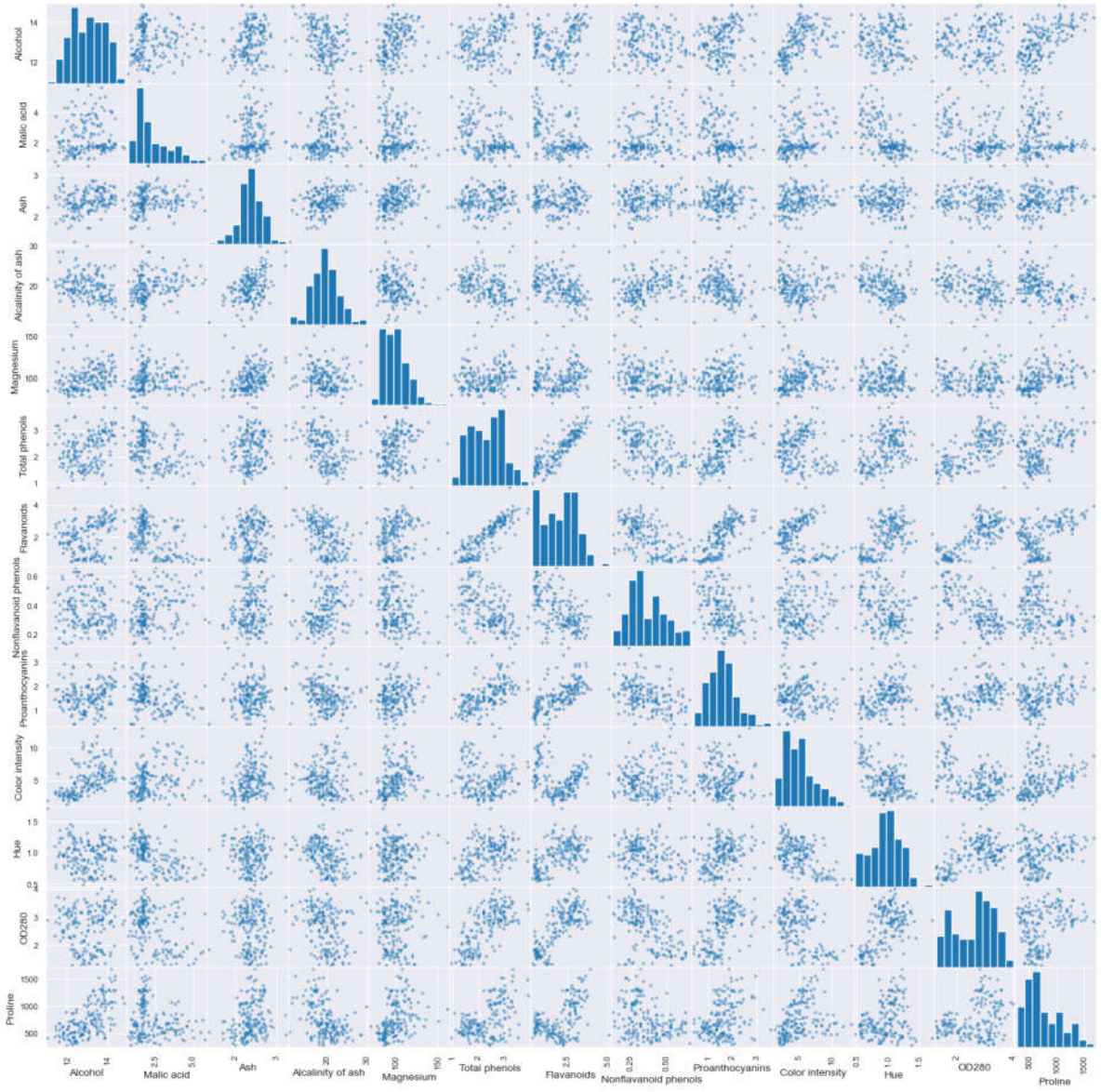
In [31]:

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

# Load the data
df = pd.read_csv("job-market.csv")

# Count the number of jobs by location
jobs_by_location = df.groupby("Location")["Title"].count()

# Sort the locations by number of jobs in descending order
jobs_by_location = jobs_by_location.sort_values(ascending=False)

# Count the number of jobs by classification within each location
jobs_by_class = df.groupby(["Location", "Classification"])["Title"].count

# Define the color scheme for the pie chart
cmap = plt.get_cmap("tab20")
labels = ['Low' , 'Medium' , 'High' , 'Critical']
# Create the pie chart
fig, ax = plt.subplots()
size = 0.3
outer_colors = cmap(np.arange(len(jobs_by_location))*2)
inner_colors = cmap(np.arange(len(jobs_by_class)))
ax.pie(jobs_by_location, radius=1, colors=outer_colors,
       wedgeprops=dict(width=size, edgecolor='w'))
ax.pie(jobs_by_class, radius=1-size, colors=inner_colors,
       wedgeprops=dict(width=size, edgecolor='w'))
ax.set(aspect="equal", title='Job Posts by salary range')
plt.show()

```



In [32]:

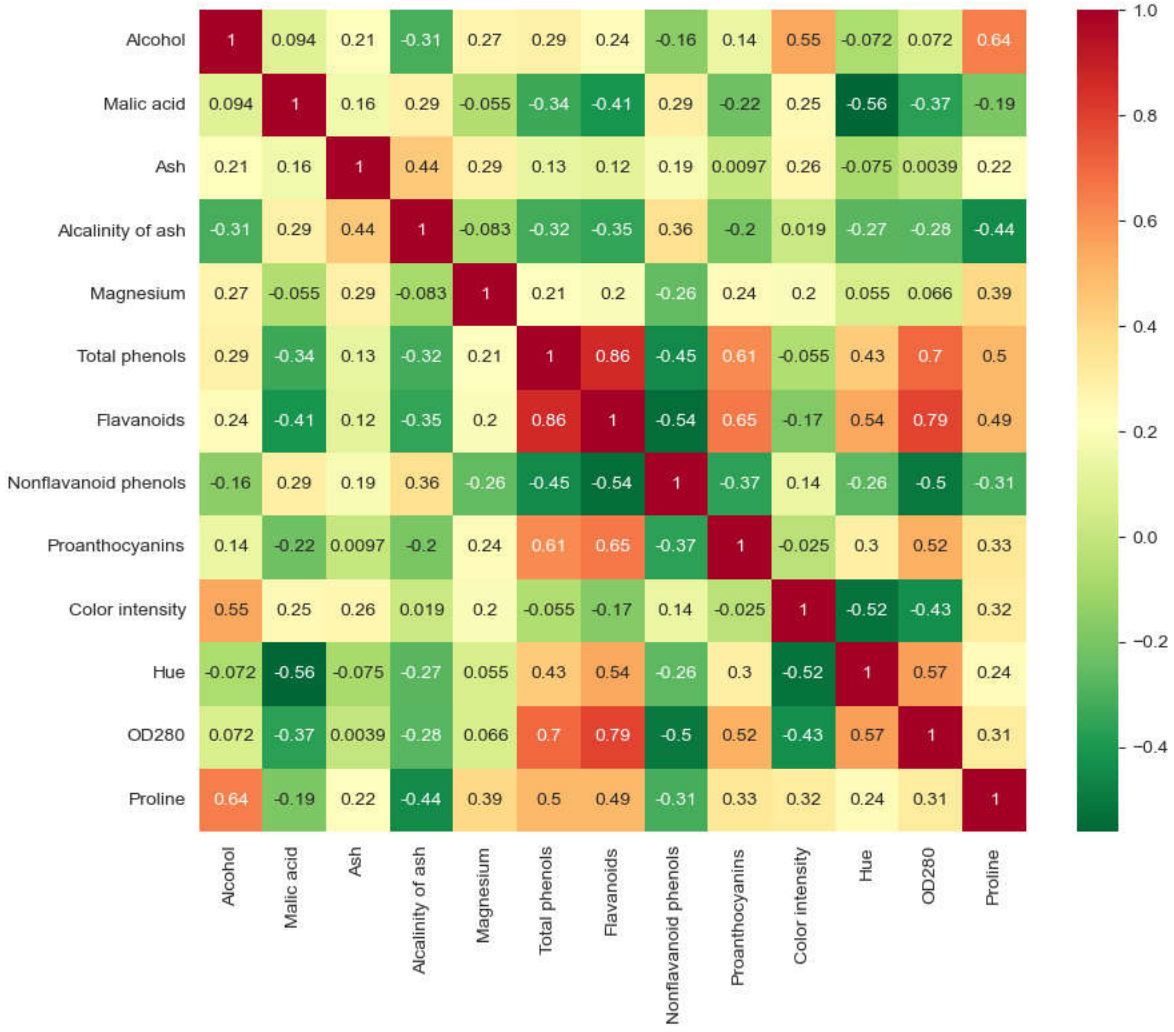
```
import pandas as pd
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix

# Load data
data = pd.read_csv('wine.data.csv')

# Selecting numerical features
cols = ['Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash',
       'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
       'Proanthocyanins', 'Color intensity', 'Hue', 'OD280', 'Proline']
data_subset = data[cols]

# Plotting the scatter matrix
scatter_matrix(data_subset, figsize=(18,18))
plt.show()

```



In [33]:
```
print(data.columns)

```

```
Index(['Label', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash',
       'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',
       'Proanthocyanins', 'Color intensity', 'Hue', 'OD280', 'Proline'],
      dtype='object')
```

In [34]:
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

df = pd.read_csv("wine.data.csv")

numerical_cols = df.select_dtypes(include=['float64', 'int64']).columns
numerical_cols = numerical_cols.drop('Label')

corr = df[numerical_cols].corr()

fig, ax = plt.subplots(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap='RdYlGn_r', ax=ax)

plt.show()
```



CALL ME FINE CUTE