7-data-science-data-visualzaton-1

April 23, 2024

1 Exercise1

2 Follow along with these steps:

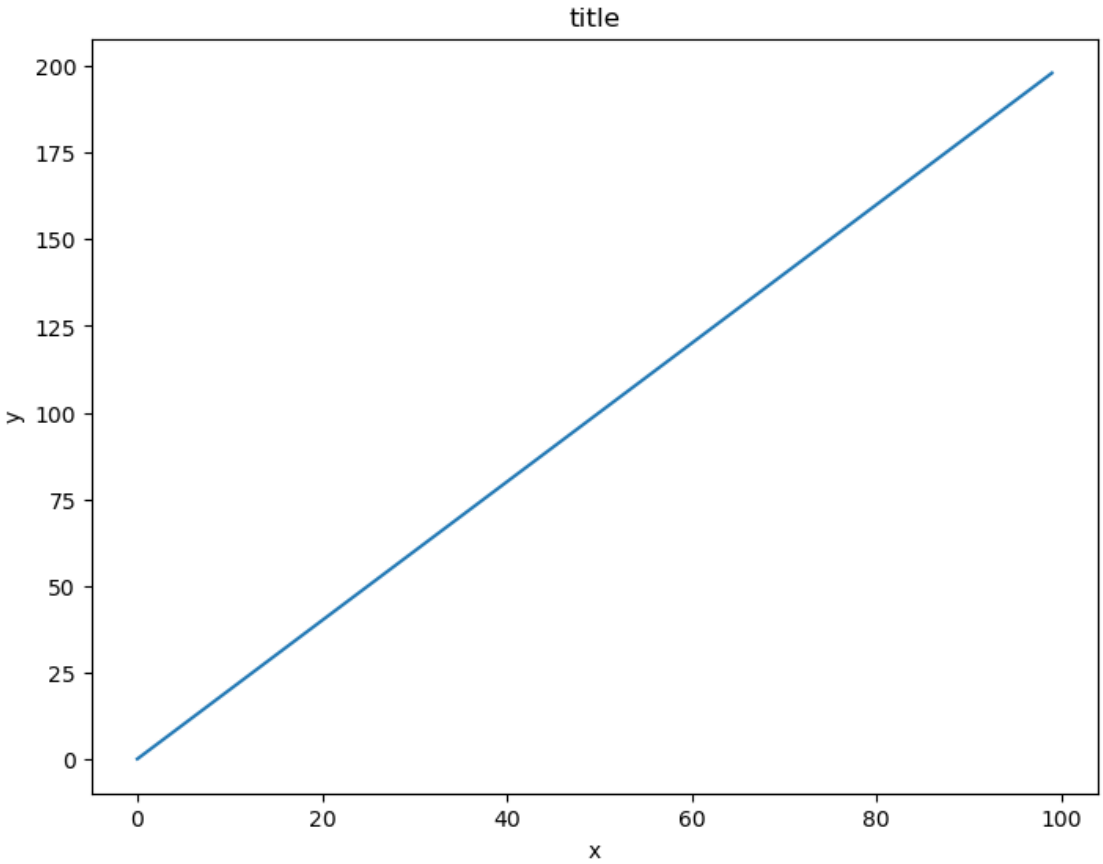
Create a figure object called fig using plt.figure() Use add_axes to add an axis to the figure canvas at [0,0,1,1]. Call this new axis ax Plot (x,y) on that axes and set the labels and titles to match the plot below:

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

[86]:
```
import numpy as np
x = np.arange(0,100)
y = x*2
z = x**2
```

[3]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.plot(x,y)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('title')
```

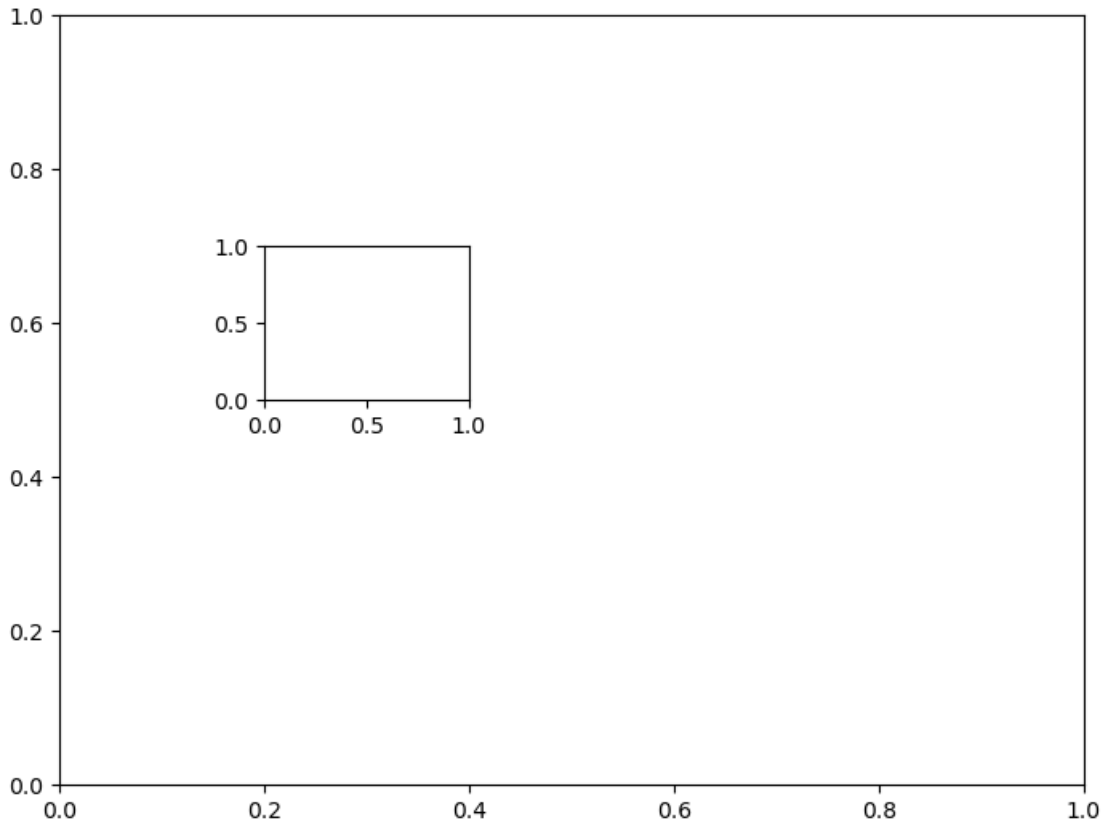
[3]:
```
Text(0.5, 1.0, 'title')
```



3 Exercise-2

Create a figure object and put two axes on it, ax1 and ax2. Located at [0,0,1,1] and [0.2,0.5,.2,.2] respectively.

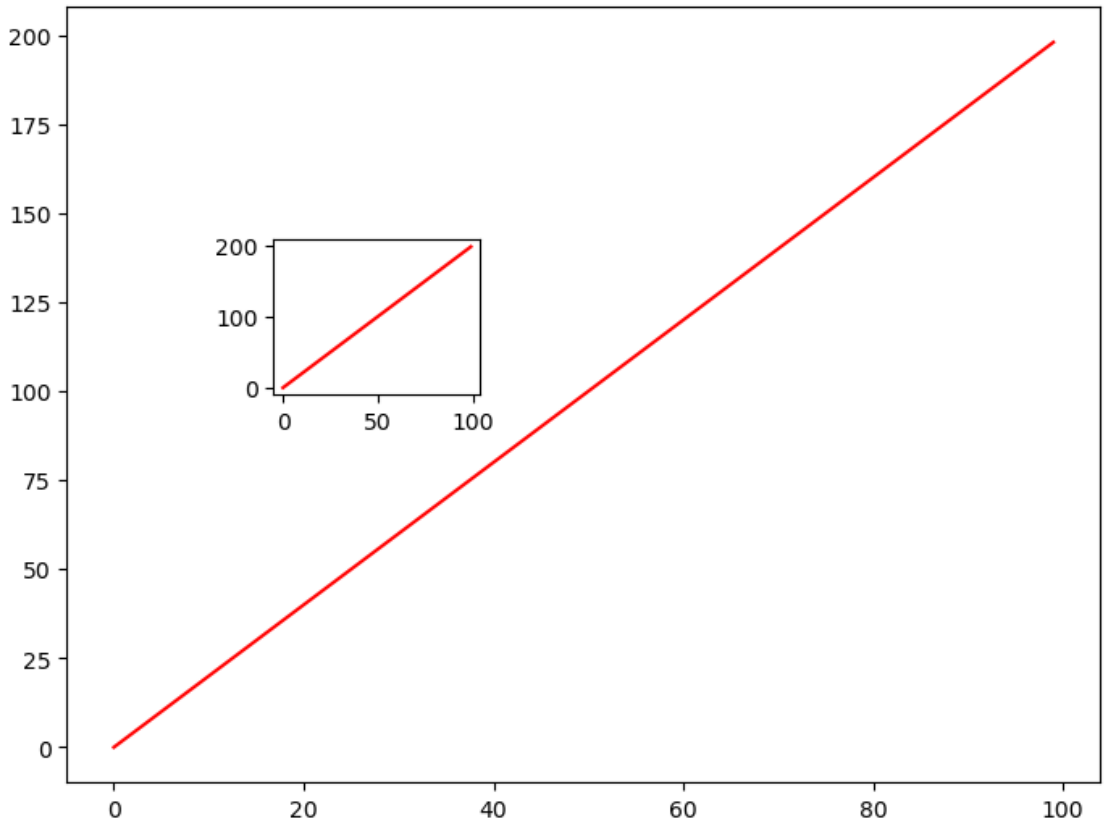
[4]:
```
fig = plt.figure()
ax1 = fig.add_axes([0,0,1,1])
ax2 = fig.add_axes([0.2,0.5,.2,.2])
```



Now plot (x,y) on both axes. And call your figure object to show it.

[5]:
```
ax1.plot(x,y,'r')
ax2.plot(x,y,'r')
fig
```

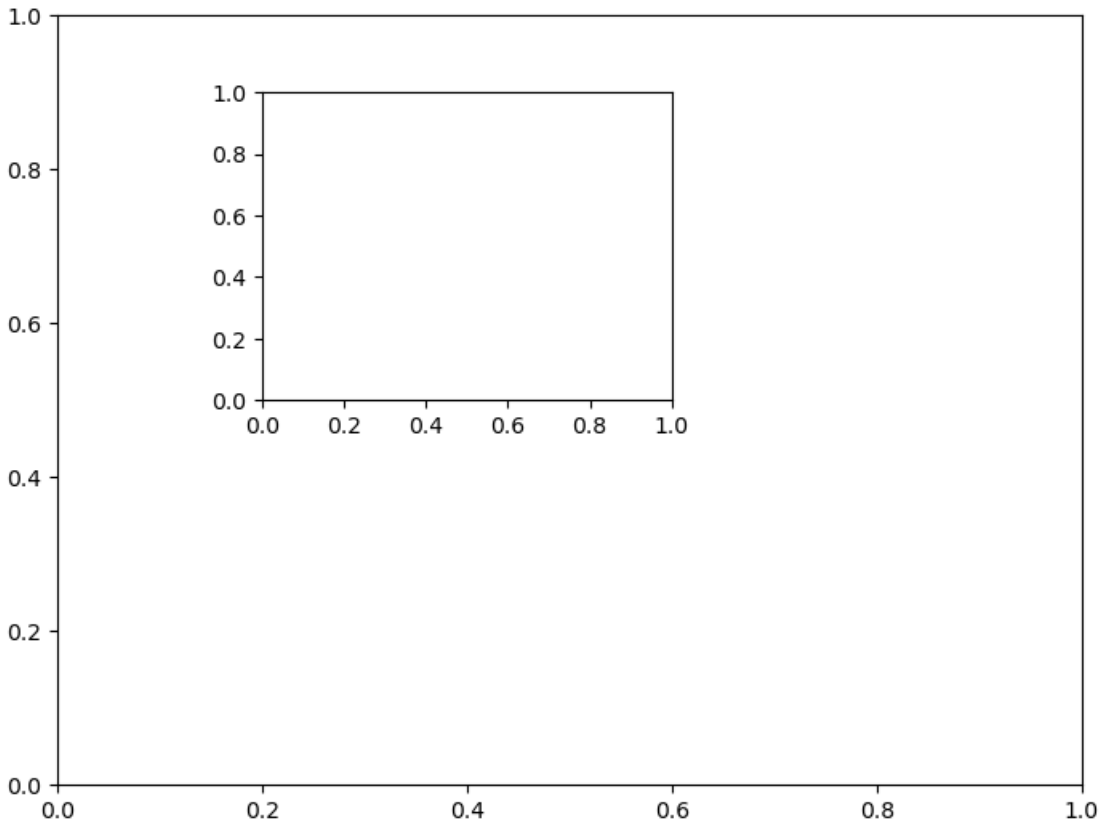
[5]:



4 Exercise 3

4.0.1 Create the plot below by adding two axes to a figure object at [0,0,1,1] and [0.2,0.5,.4,.4]

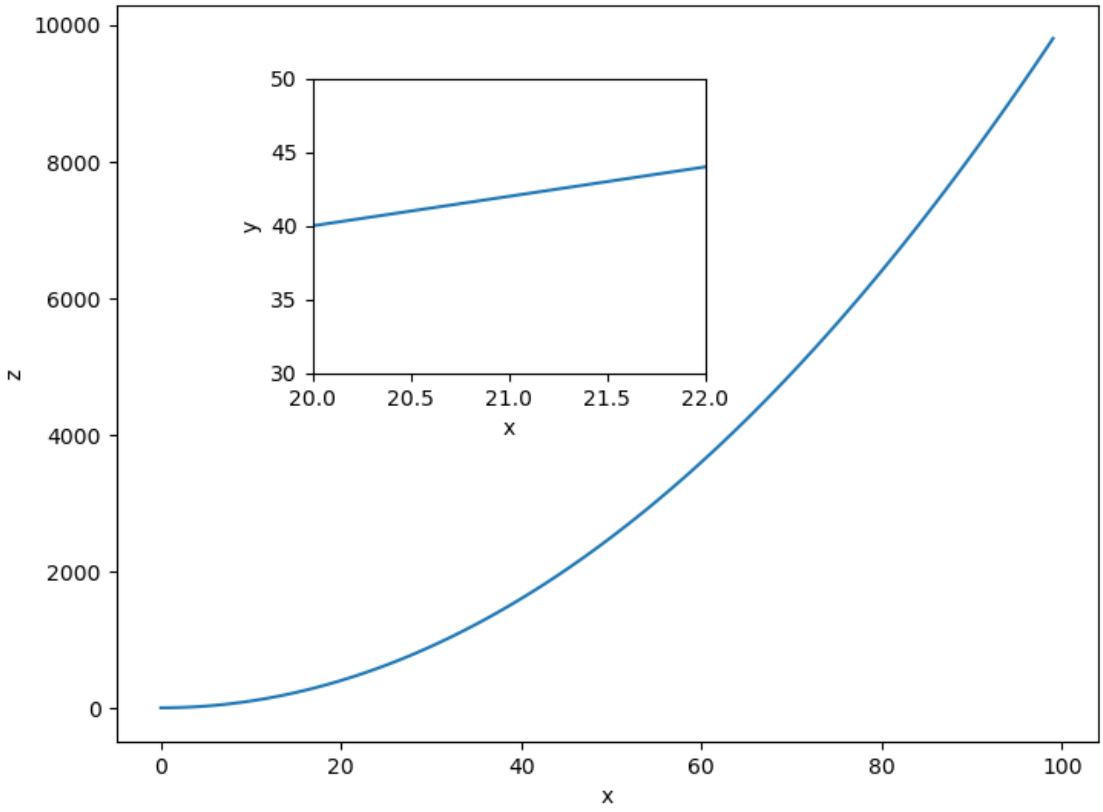
[6]:
```
fig1 = plt.figure()
ax1 = fig1.add_axes([0,0,1,1])
ax2 = fig1.add_axes([0.2,0.5,.4,.4])
```



4.0.2 Now use x,y, and z arrays to recreate the plot below. Notice the xlimits and y limits on the inserted plot:

[7]:
```
ax1.plot(x,z)
ax1.set_xlabel('x')
ax1.set_ylabel('z')
ax2.plot(x,y)
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.set_xlim([20,22])
ax2.set_ylim([30,50])
fig1
```

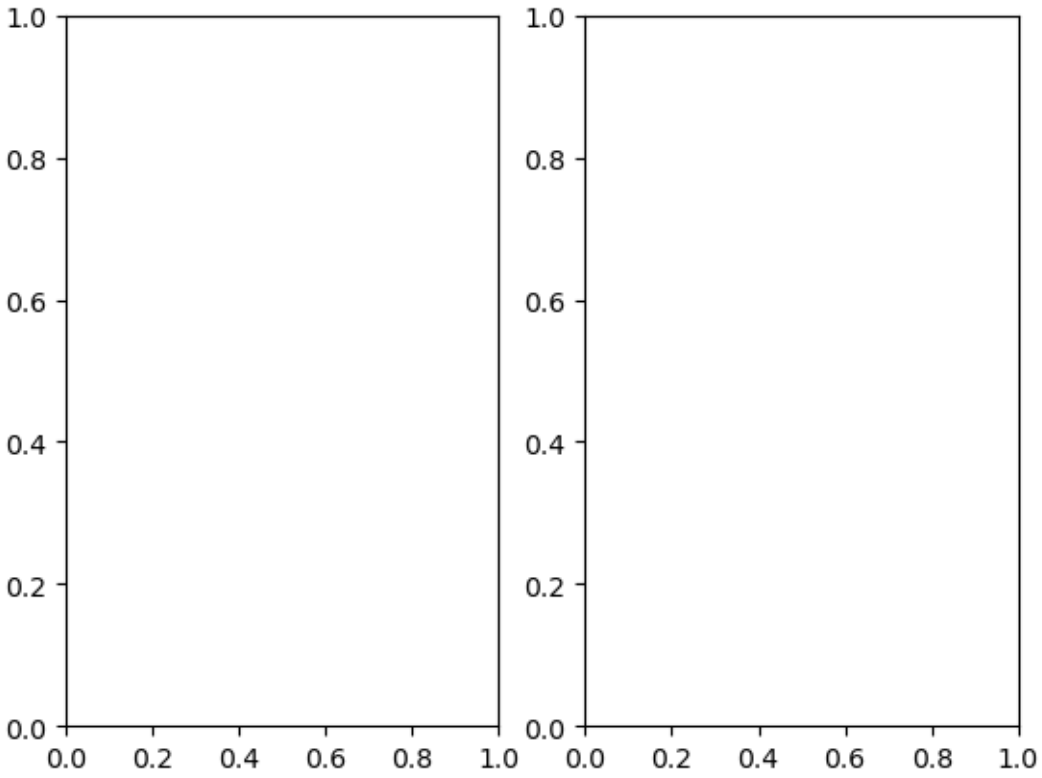
[7]:



5 Exercise 4

Use plt.subplots(nrows=1, ncols=2) to create the plot below.

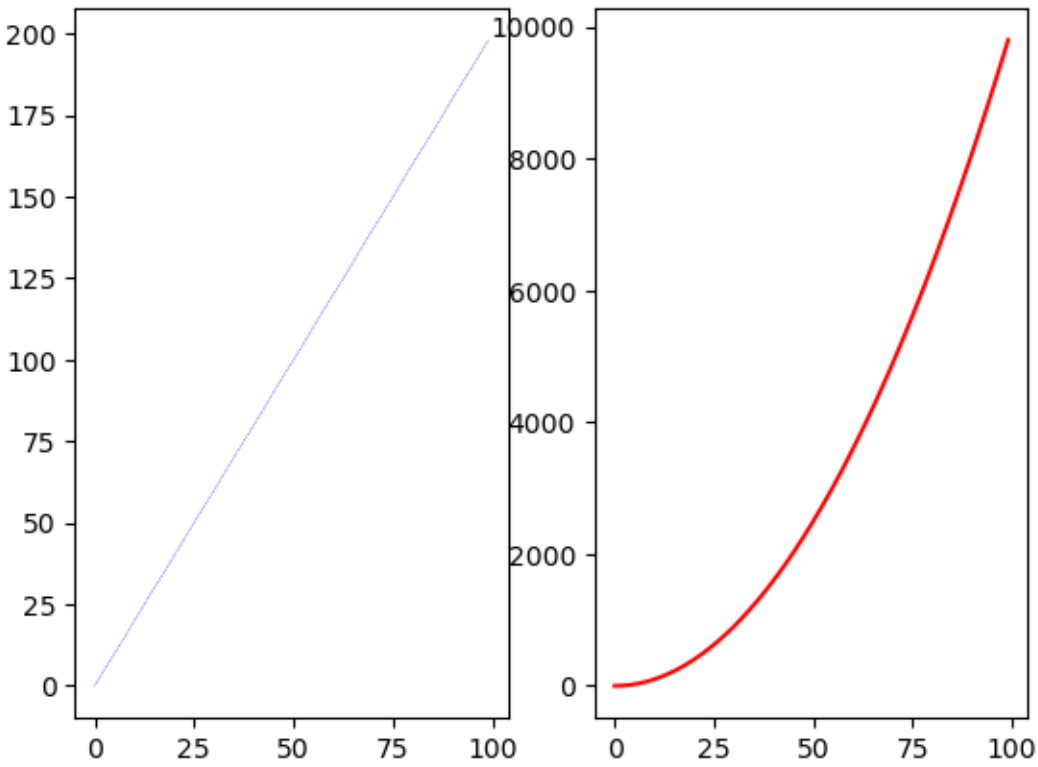
[8]:
```
fig, axes = plt.subplots(nrows=1, ncols=2)
```



Now plot (x,y) and (x,z) on the axes. Play around with the linewidth and style

[9]:
```
axes[0].plot(x,y,linestyle='--',color='blue',linewidth=0.25)
axes[1].plot(x,z,linestyle='-',color='red')
fig
```

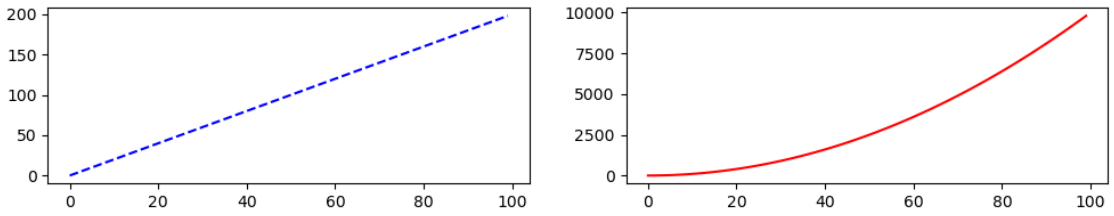
[9]:

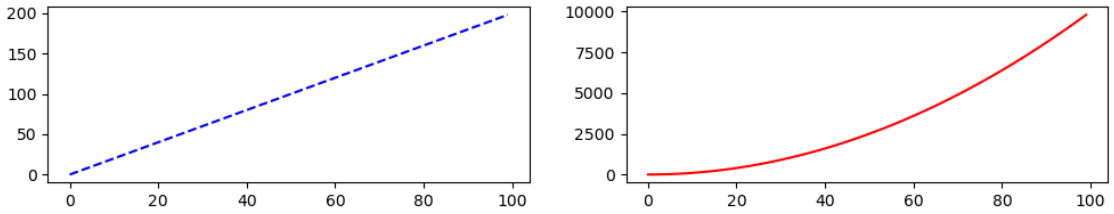


See if you can resize the plot by adding the figsize() argument in plt.subplots() are copying and pasting your previous code.

[10]:
```
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(12,2))
axes[0].plot(x,y,'b--')
axes[1].plot(x,z,'r-')
fig
```

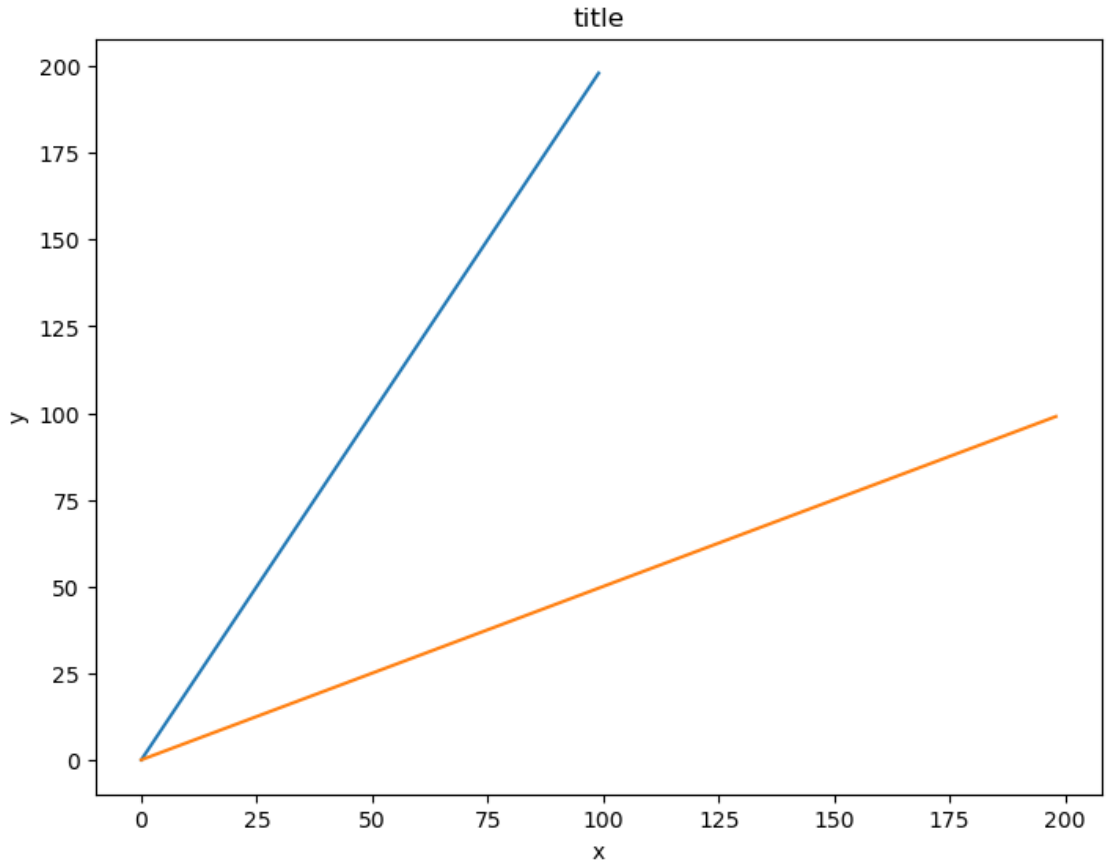
[10]:



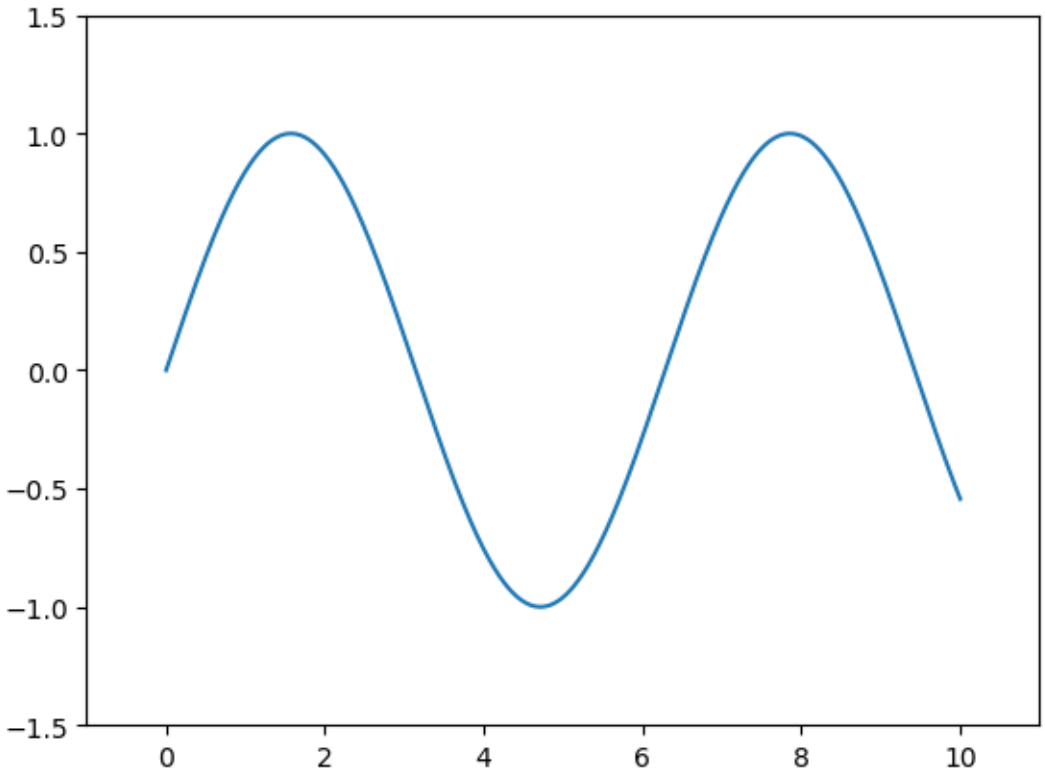


[11]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.plot(x,y)
ax.plot(y,x)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('title')
```

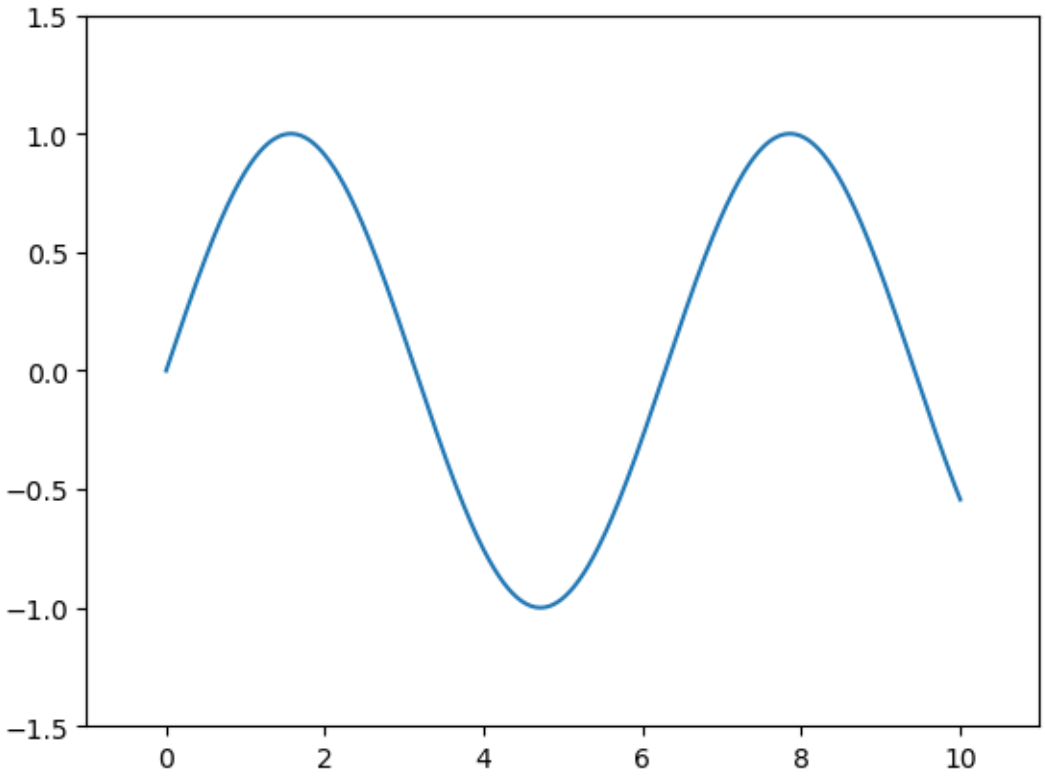
[11]: Text(0.5, 1.0, 'title')



[12]:
```
x = np.linspace(0, 10, 1000)
plt.plot(x, np.sin(x))
plt.xlim(-1, 11)
plt.ylim(-1.5, 1.5);
```

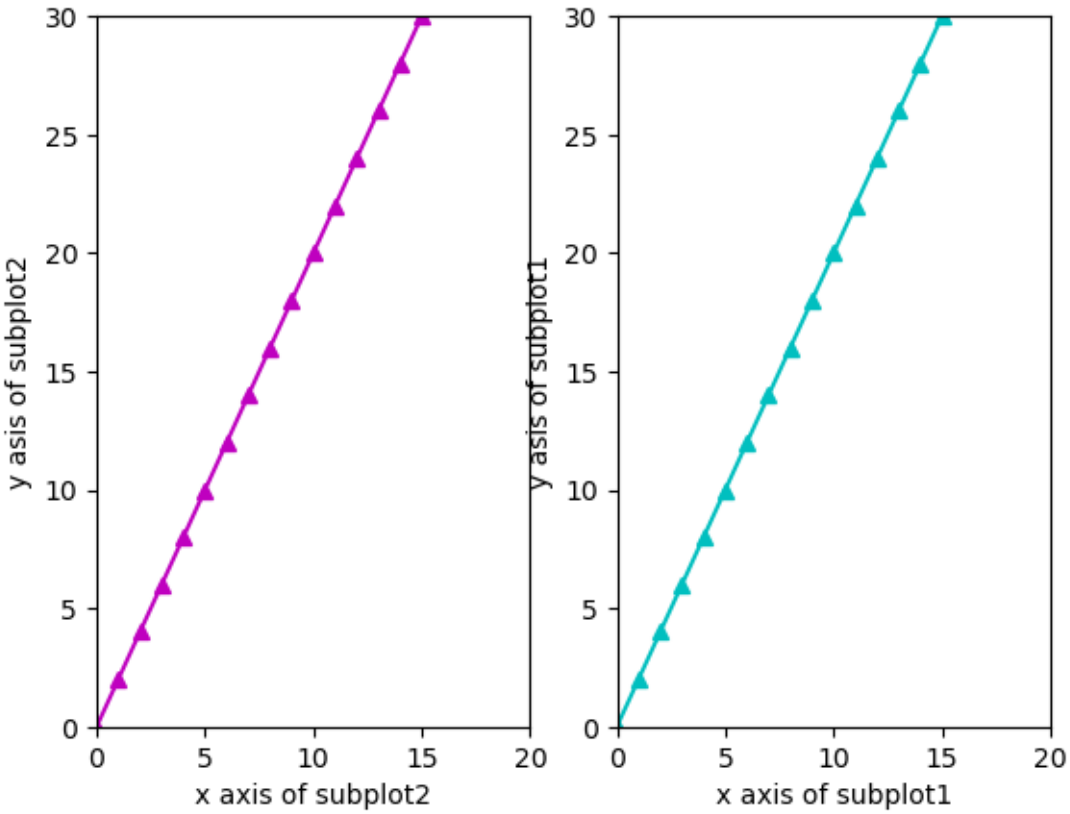


[13]:
```
plt.plot(x, np.sin(x))
plt.axis([-1, 11, -1.5, 1.5]);
```



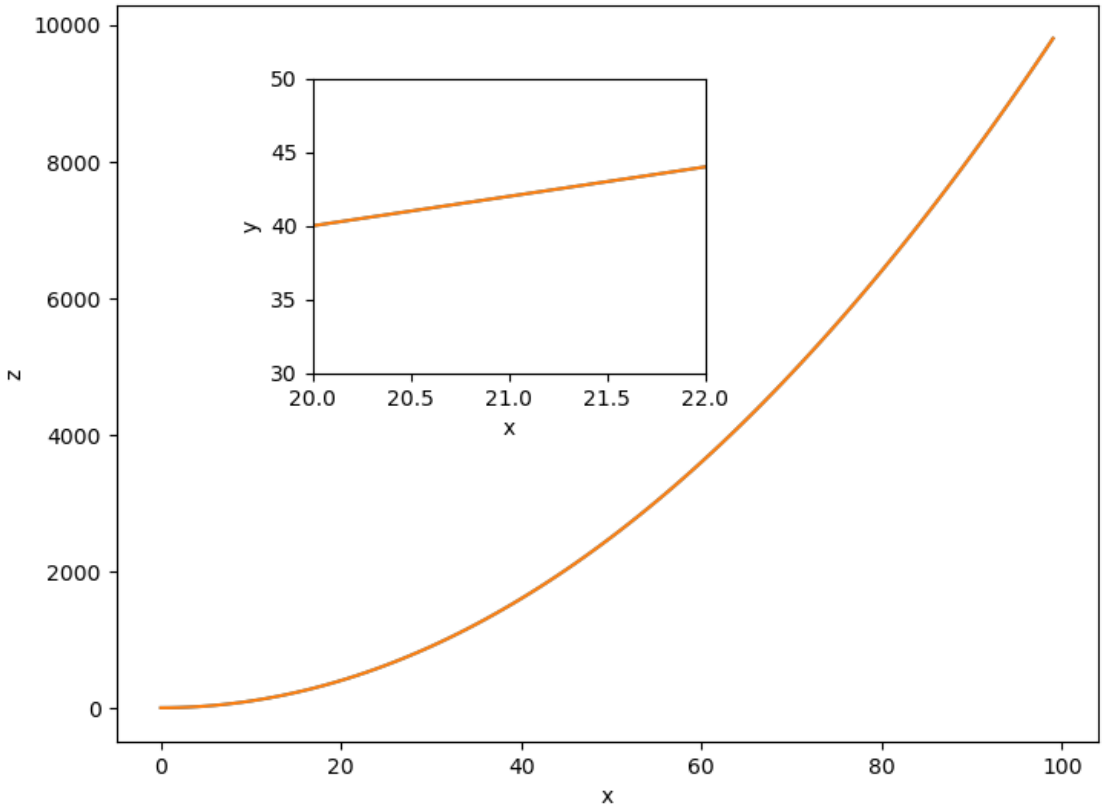
[16]:
```
plt.subplot(1,2,1)#row,col,plotn
plt.plot(x,y,'m^-')
plt.xlabel('x axis of subplot2')
plt.ylabel('y asis of subplot2')
plt.xlim(0,20)
plt.ylim(0,30)


plt.subplot(1,2,2)#row,col,plotn
plt.plot(x,y,'c^-')
plt.xlabel('x axis of subplot1')
plt.ylabel('y asis of subplot1')
plt.xlim(0,20)
plt.ylim(0,30)
```

[16]:
```
(0.0, 30.0)
```

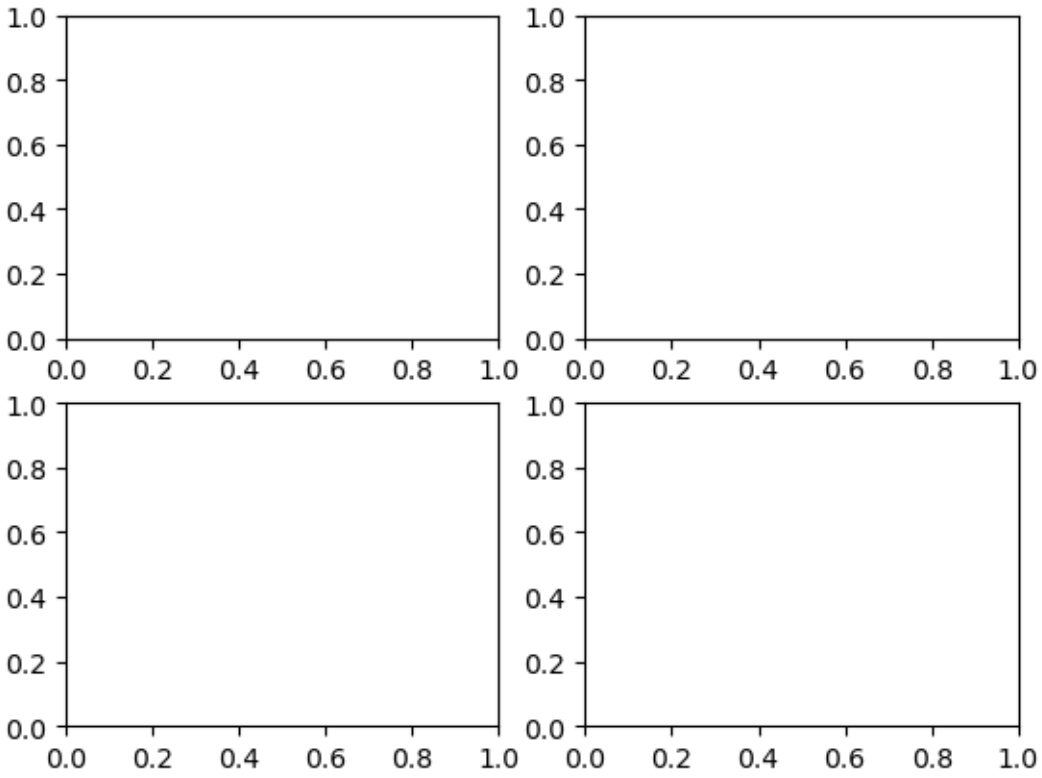


[17]:
```
ax1.plot(x,z)
ax1.set_xlabel('x')
ax1.set_ylabel('z')
ax2.plot(x,y)
ax2.set_xlabel('x')
ax2.set_ylabel('y')
ax2.set_xlim([20,22])
ax2.set_ylim([30,50])
fig1
```

[17]:

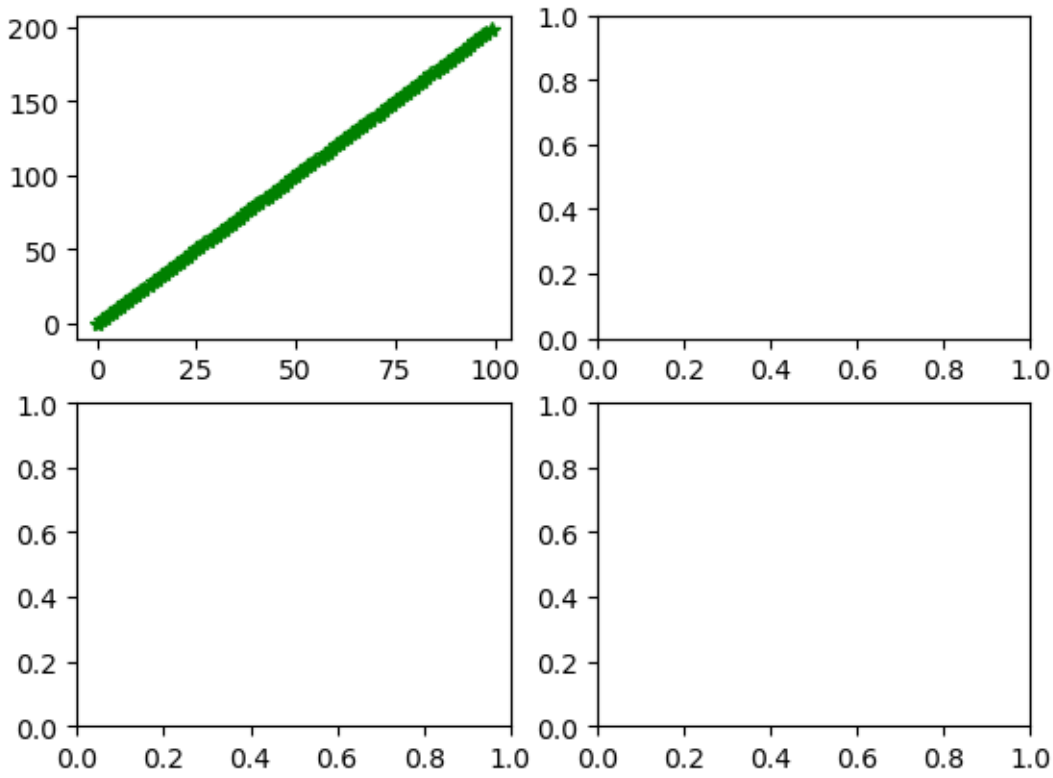


[18]:
```
fig, axes = plt.subplots(nrows=2, ncols=2)
```

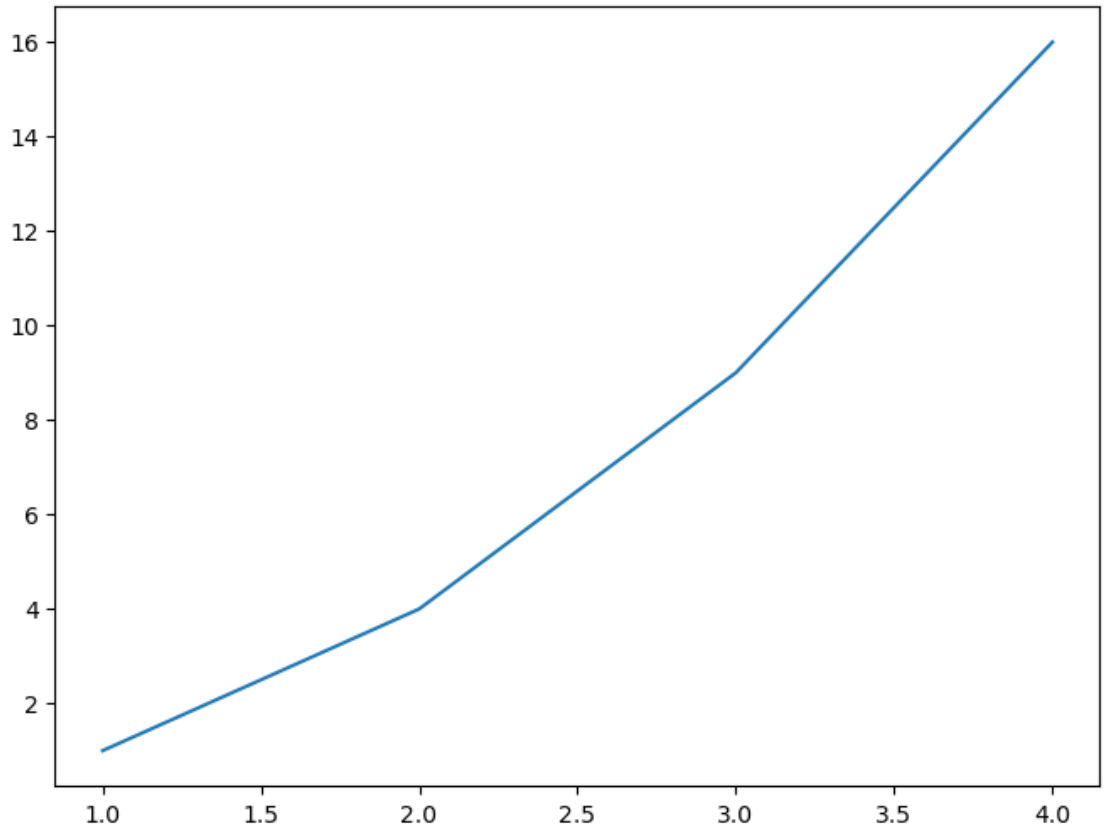


[19]:
```
#Axes is an array of axes to plot on
axes[0,0].plot(x,y,'g*-')
fig
```

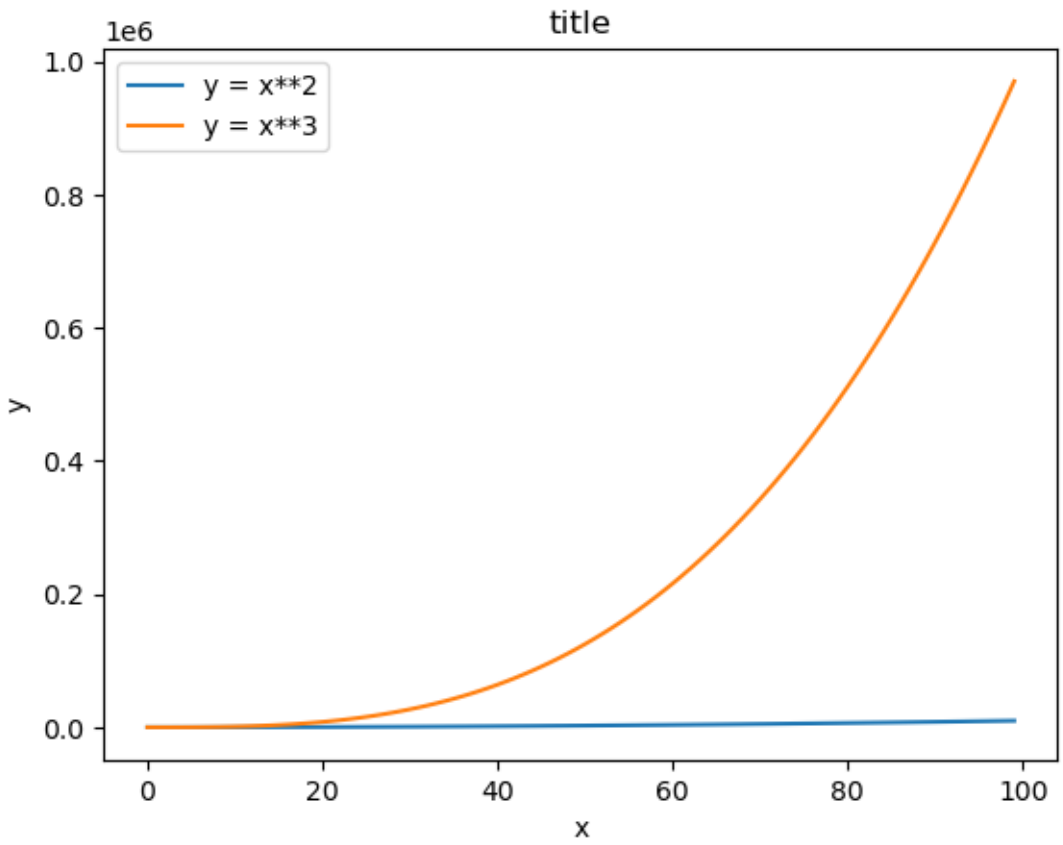
[19]:



[20]:
```
# Create a figure and plot some data
plt.figure(figsize=(8, 6))
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])

# Save the figure with a specific DPI
plt.savefig('my_plot.png', dpi=300)
```



[21]:
```
fig, ax = plt.subplots()

ax.plot(x, x**2, label="y = x**2")
ax.plot(x, x**3, label="y = x**3")
ax.legend(loc=2); # upper left corner
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('title');
```

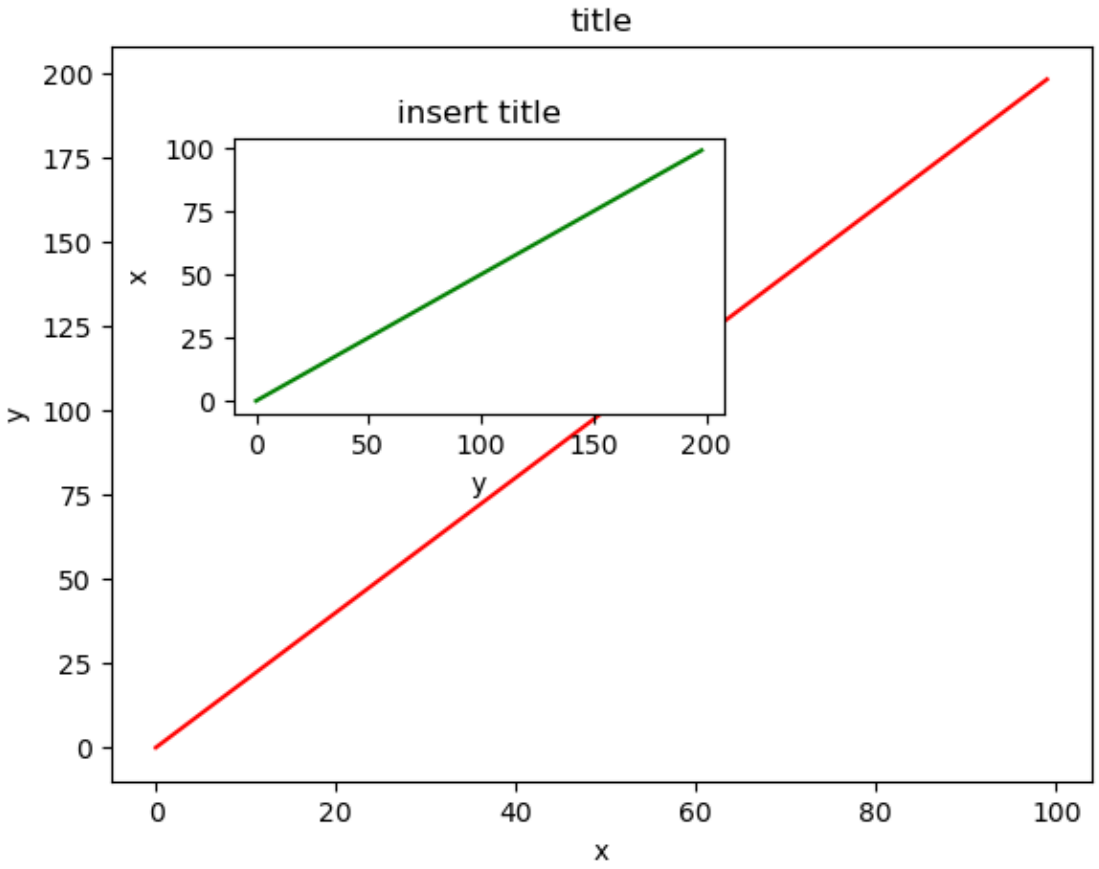


6 Introduction to the Object Oriented Method

The main idea of using the more formal Object Oriented method is to create figure objects and then just call methods or attributes off of the

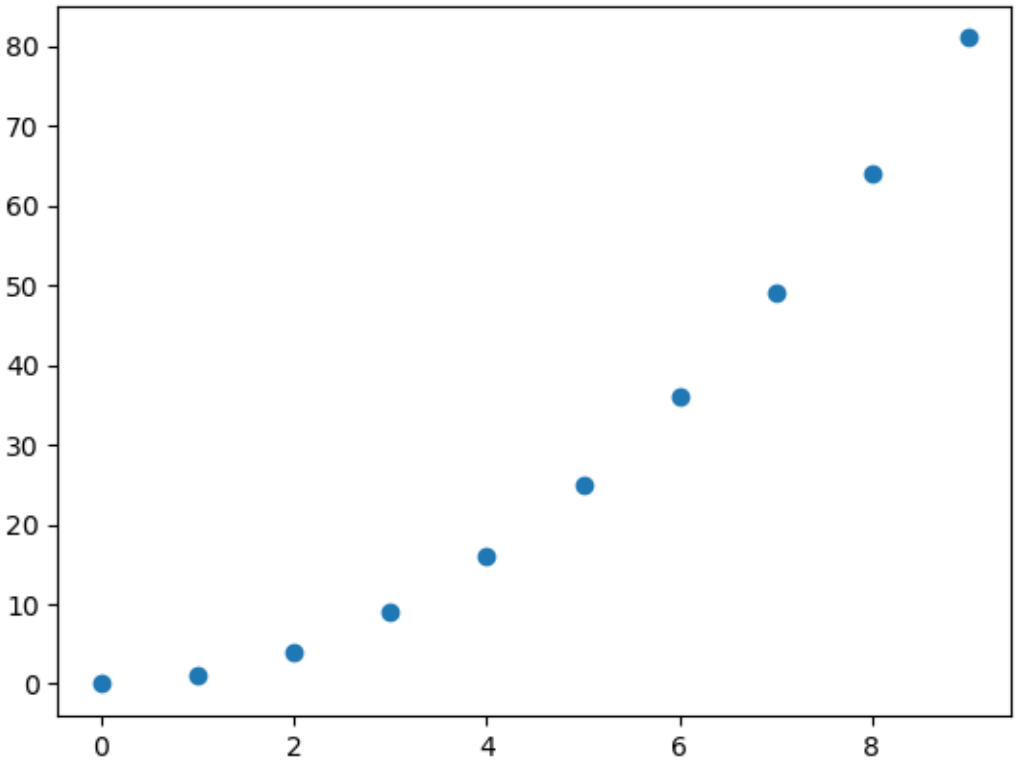
[22]:
```
fig = plt.figure()

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

# main figure
axes1.plot(x, y, 'r')
axes1.set_xlabel('x')
axes1.set_ylabel('y')
axes1.set_title('title')

# insert
axes2.plot(y, x, 'g')
axes2.set_xlabel('y')
```

```
axes2.set_ylabel('x')
axes2.set_title('insert title');
```



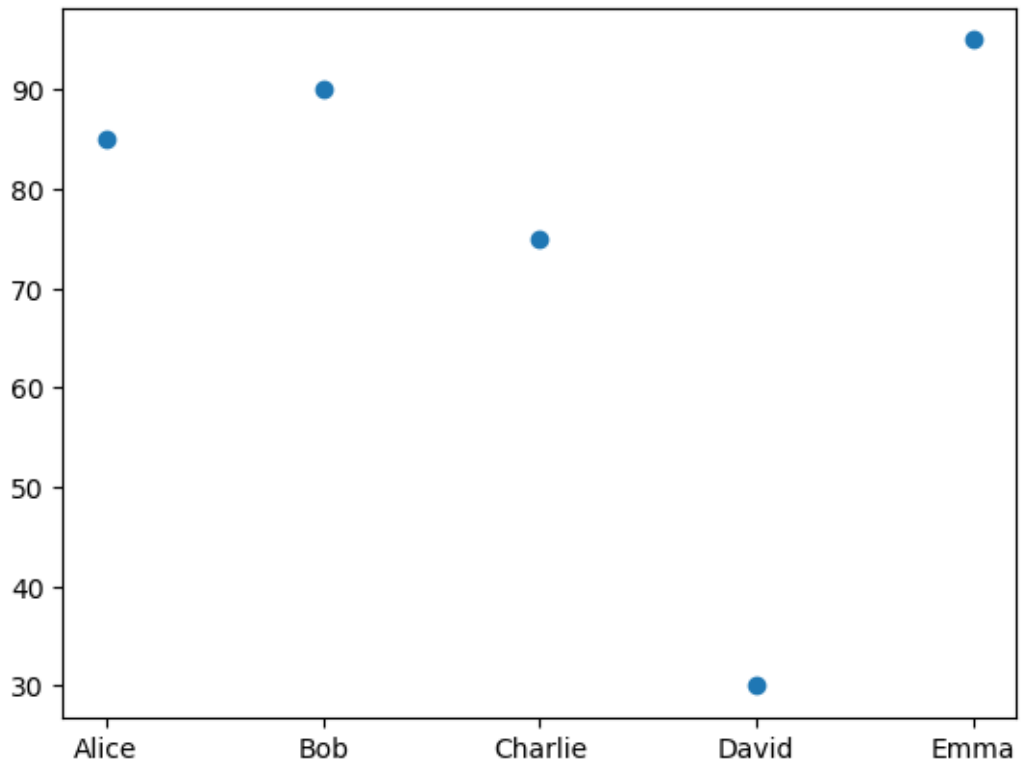
[23]:
```
a=np.arange(10)
b=a**2
plt.scatter(a,b)
```

[23]: <matplotlib.collections.PathCollection at 0x1dc5311bcd0>



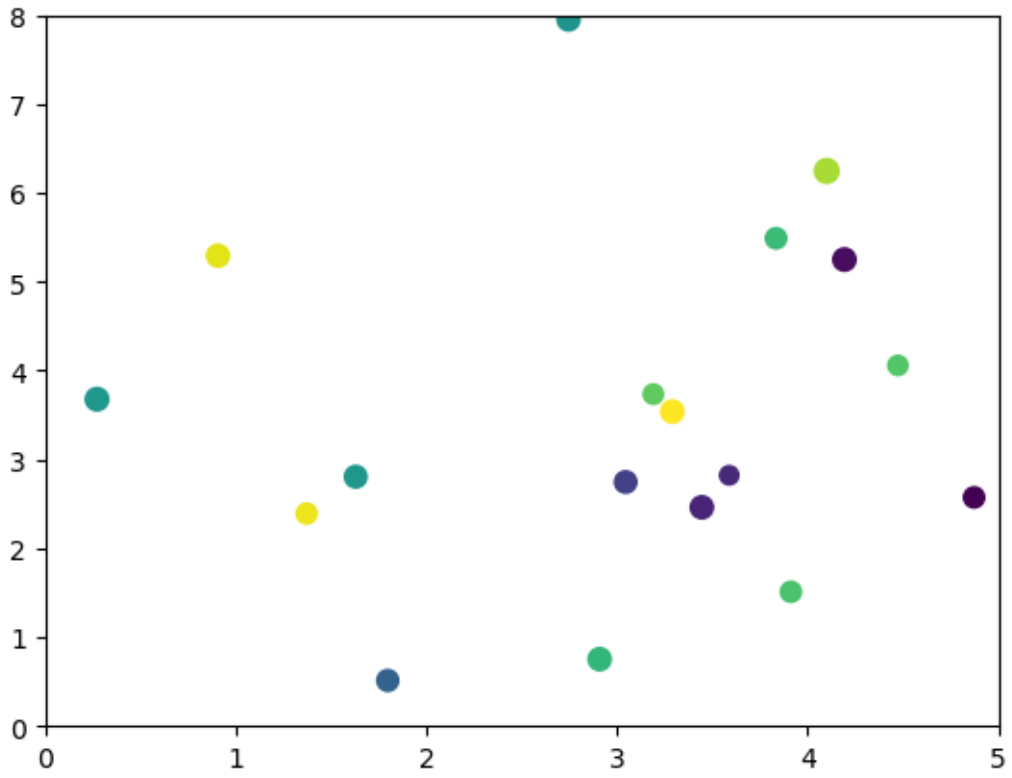
[24]:
```
import pandas as pd
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emma'],
    'Python': [85, 90, 75, 30, 95],
    'Science': [70, 80, 85, 100, 95],
    'History': [75, 85, 80, 70, 90],
    'English': [80, 75, 90, 85, 95]
}
df = pd.DataFrame(data)
plt.scatter(df.Name,df.Python)
```

[24]:
```
<matplotlib.collections.PathCollection at 0x1dc544222d0>
```



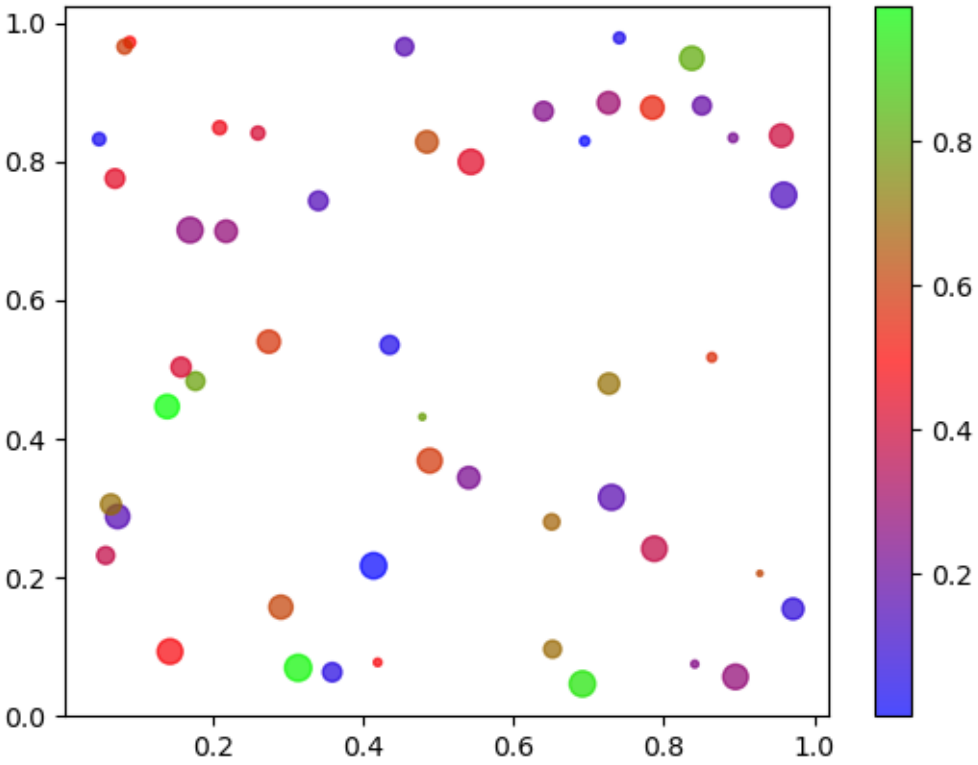
[25]:
```
np.random.seed(3)
x=4+np.random.normal(0,2,24)
y=4+np.random.normal(0,2,len(x))
#size and color:
sizes=np.random.uniform(45,80,len(x))
colors=np.random.uniform(15,80,len(x))
#plot
fig,ax=plt.subplots()

ax.scatter(x,y,s=sizes,c=colors)
ax.set(xlim=(0,5),ylim=(0,8))
```

[25]:
```
[(0.0, 5.0), (0.0, 8.0)]
```

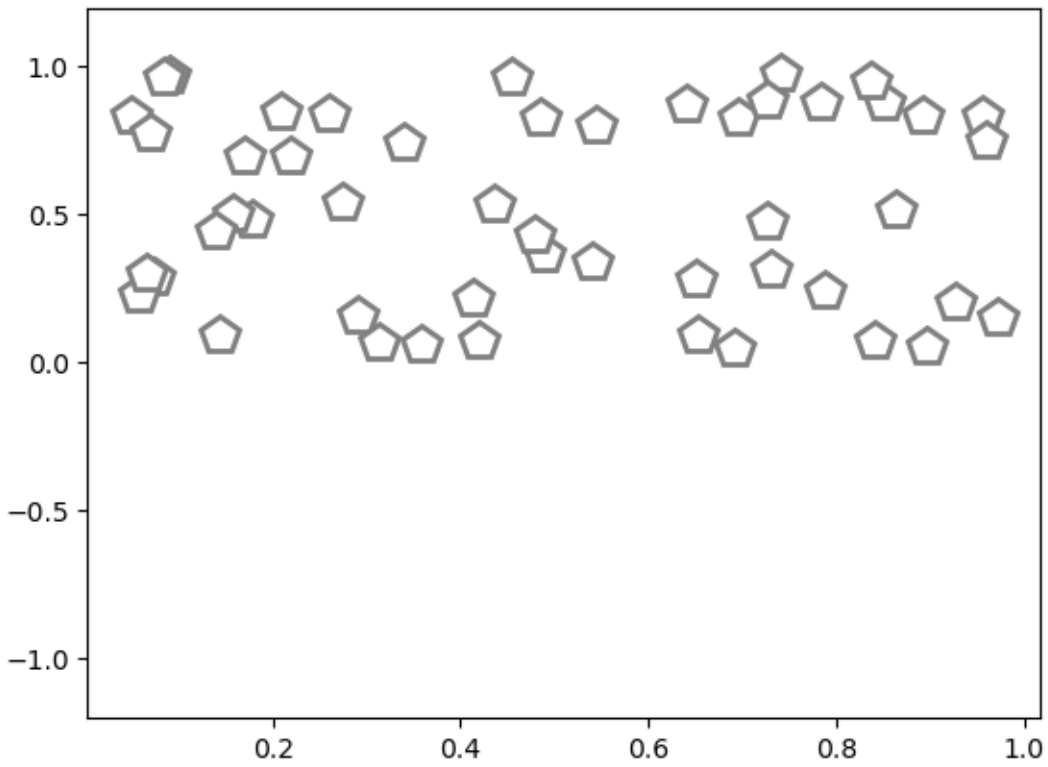


[26]:
```
x=np.random.rand(50)
y=np.random.rand(50)

colors=np.random.rand(50)
sixes=100*np.random.rand(50)

plt.scatter(x,y,c=colors,s=sixes,alpha=0.7,cmap='brg')
plt.colorbar()
```

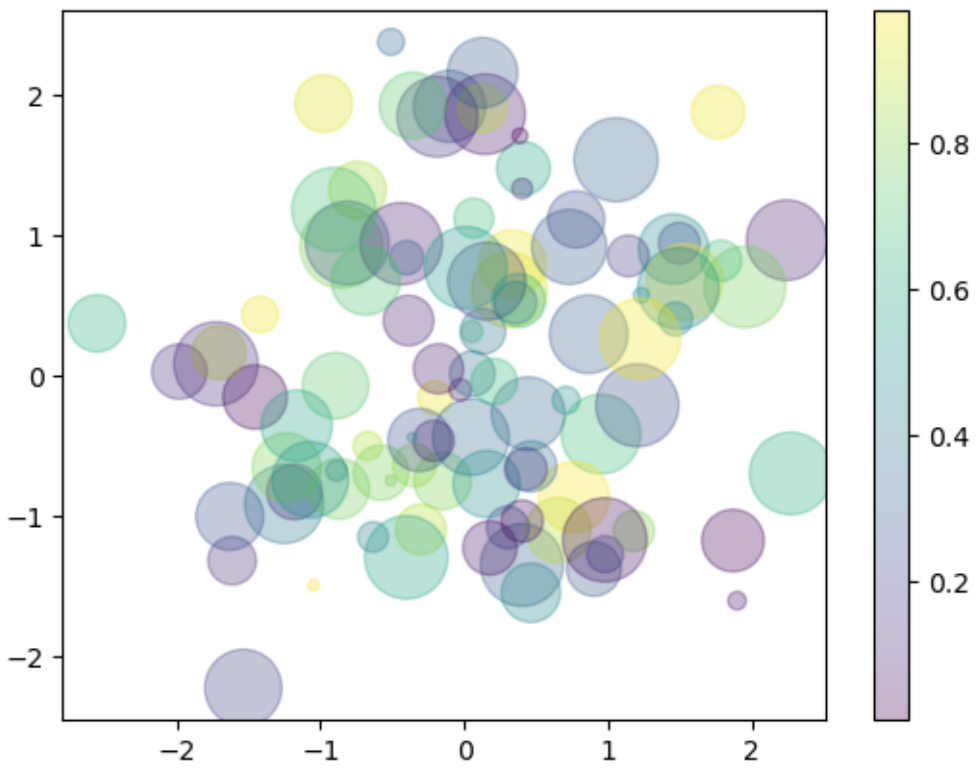
[26]:
```
<matplotlib.colorbar.Colorbar at 0x1dc4c890710>
```



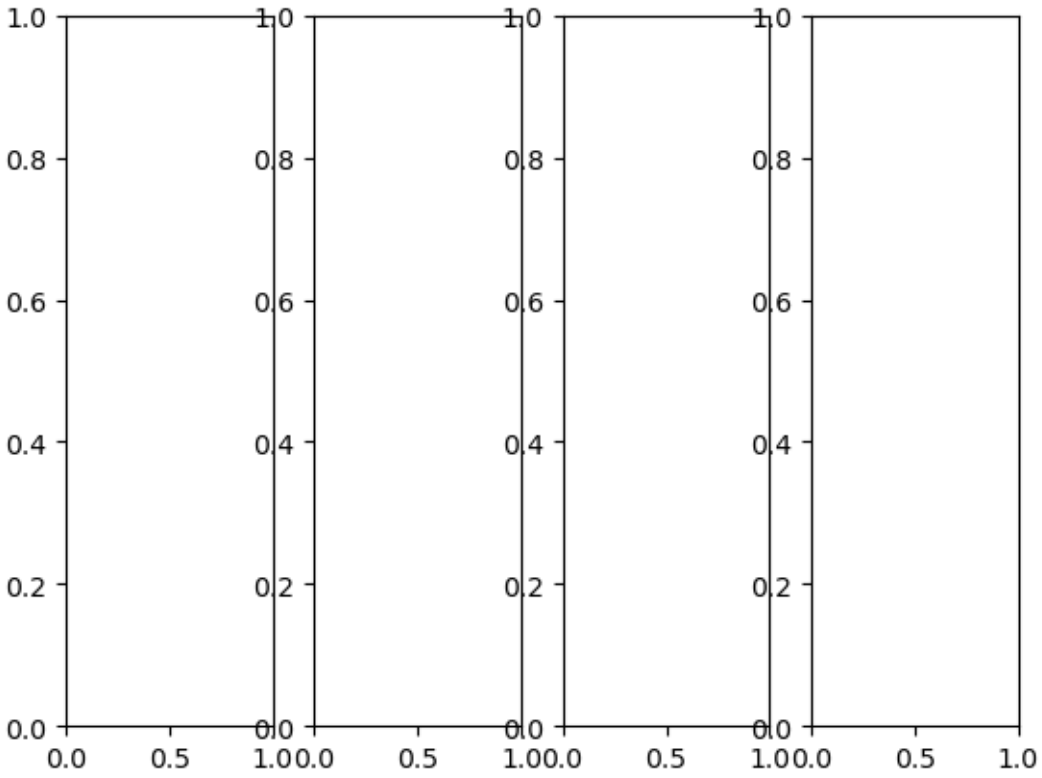
[27]:
```
plt.plot(x, y, 'p', color='gray',
markersize=15, linewidth=4,
markerfacecolor='white',
markeredgecolor='gray',
markeredgewidth=2)
plt.ylim(-1.2, 1.2);
```



[28]:
```
rng = np.random.RandomState(0)
x = rng.randn(100)
y = rng.randn(100)
colors = rng.rand(100)
sizes = 1000 * rng.rand(100)
plt.scatter(x, y, c=colors, s=sizes, alpha=0.3,
cmap='viridis')
plt.colorbar(); # show color scale
```

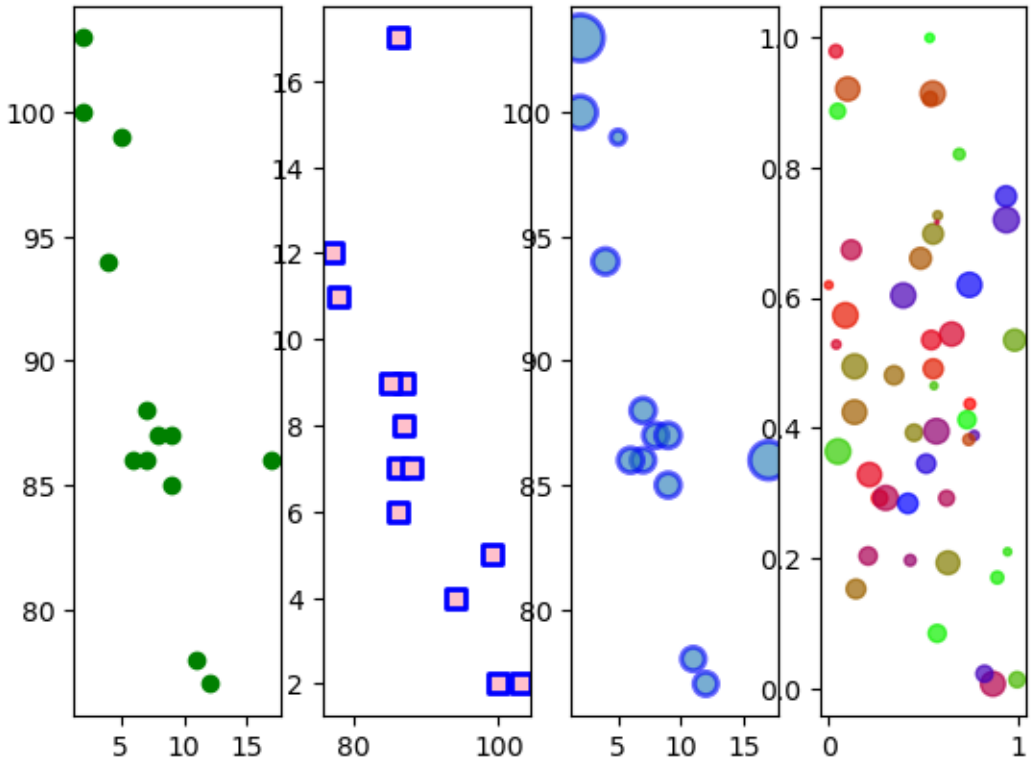


[29]:
```
fig, ax = plt.subplots(1, 4)
```

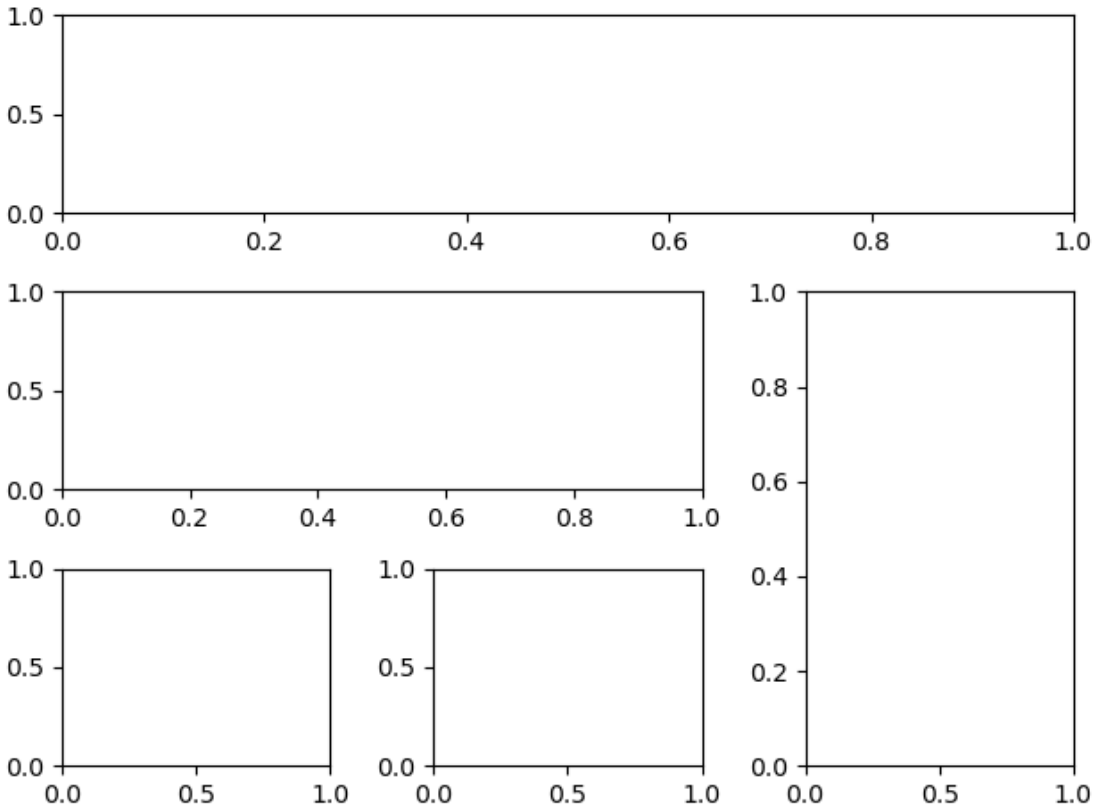


[30]:
```
x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
y = [99, 86, 87, 88, 100, 86, 103, 87, 94, 78, 77, 85, 86]
bubble_sizes = [30, 80, 85, 86, 150, 200, 300, 87, 94, 78, 77, 85, 86]

fig, ax = plt.subplots(1, 4)

# Scatter plot 1
ax[0].scatter(x, y, c="green")

# Scatter plot 2
ax[1].scatter(y, x, c="pink", linewidths=2, marker="s", edgecolor="blue", s=50)

# Scatter plot 3
ax[2].scatter(x, y, s=bubble_sizes, alpha=0.6, edgecolors='b', linewidths=2)

# Scatter plot 4
x1 = np.random.rand(50)
y1 = np.random.rand(50)
colors = np.random.rand(50)
sizes = 100 * np.random.rand(50)
ax[3].scatter(x1, y1, c=colors, s=sizes, alpha=0.7, cmap="brg")
```

```
# Show all plots
plt.show()
```

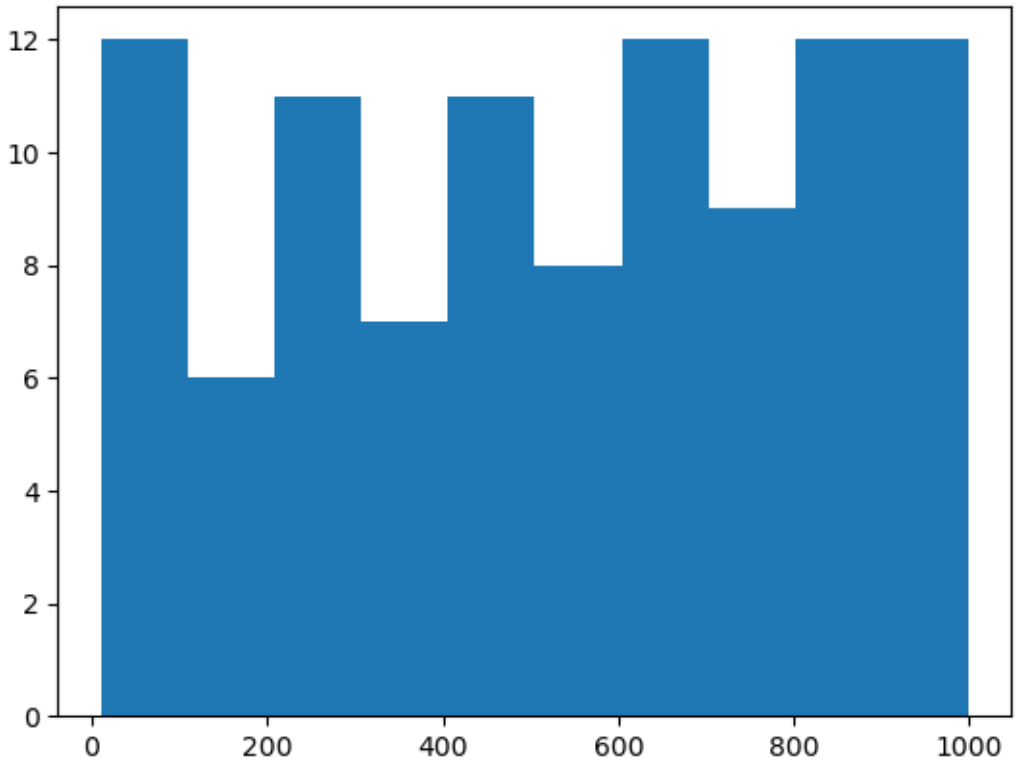


[31]:
```
fig = plt.figure()
ax1 = plt.subplot2grid((3,3), (0,0), colspan=3)
ax2 = plt.subplot2grid((3,3), (1,0), colspan=2)
ax3 = plt.subplot2grid((3,3), (1,2), rowspan=2)
ax4 = plt.subplot2grid((3,3), (2,0))
ax5 = plt.subplot2grid((3,3), (2,1))
fig.tight_layout()
```



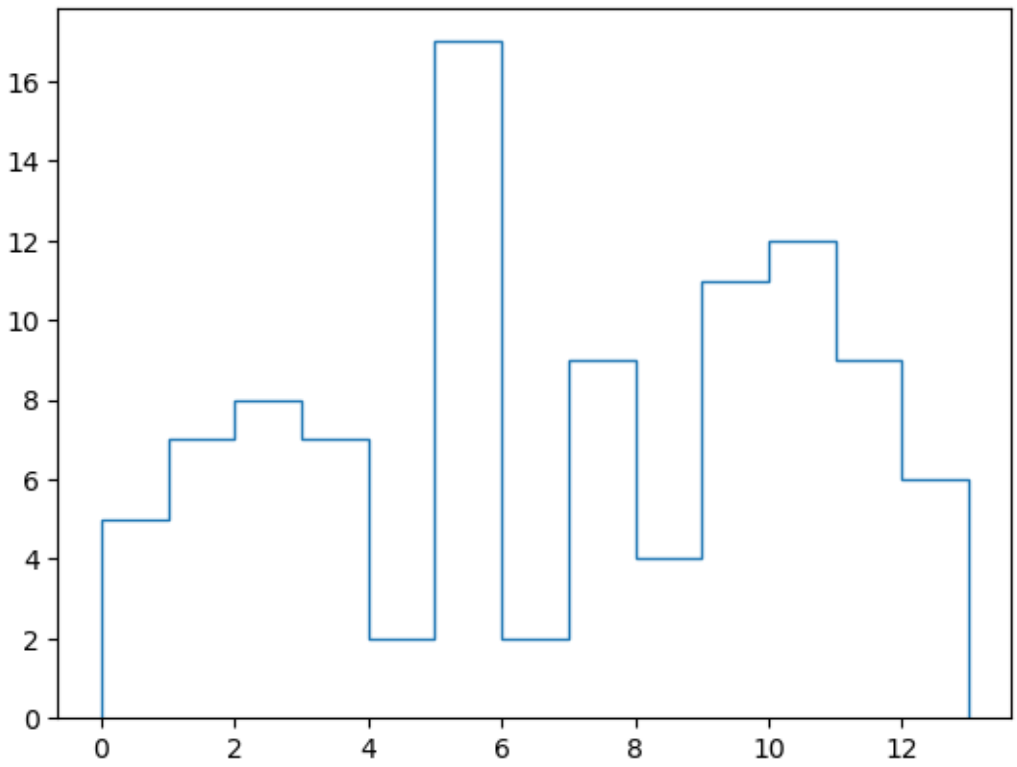
[32]:
```
from random import sample
data=sample(range(1,1000),100)
plt.hist(data)
```

[32]:
```
(array([12.,  6., 11.,  7., 11.,  8., 12.,  9., 12., 12.]),
 array([ 11. , 109.8, 208.6, 307.4, 406.2, 505. , 603.8, 702.6, 801.4,
        900.2, 999. ]),
 <BarContainer object of 10 artists>)
```



[33]:
```
plt.stairs(x)
```

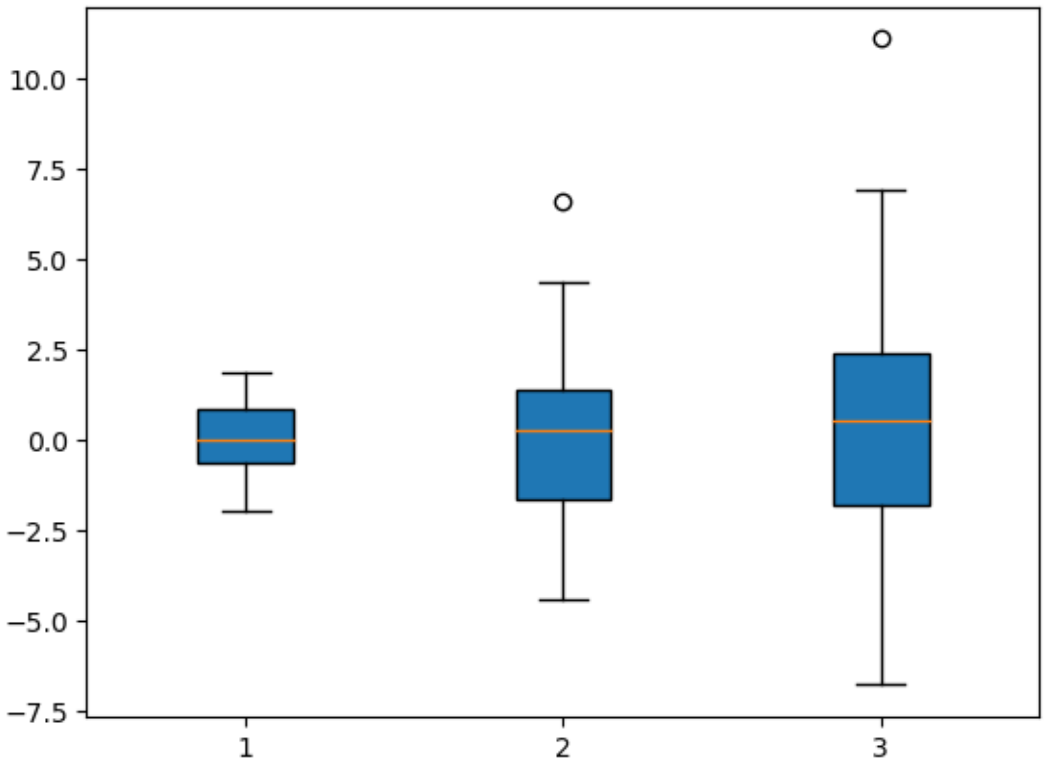
[33]:
```
<matplotlib.patches.StepPatch at 0x1dc55b0a9d0>
```



[34]:
```
data=[np.random.normal(0,std,100) for std in range(1,4)]
#rectanglar box plot
plt.boxplot(data,vert=True,patch_artist=True)
```

[34]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1dc55b23c90>,
  <matplotlib.lines.Line2D at 0x1dc55d69150>,
  <matplotlib.lines.Line2D at 0x1dc55d81550>,
  <matplotlib.lines.Line2D at 0x1dc55d80450>,
  <matplotlib.lines.Line2D at 0x1dc55d85dd0>,
  <matplotlib.lines.Line2D at 0x1dc55d86690>],
 'caps': [<matplotlib.lines.Line2D at 0x1dc55d69c90>,
  <matplotlib.lines.Line2D at 0x1dc55d6a810>,
  <matplotlib.lines.Line2D at 0x1dc55d82790>,
  <matplotlib.lines.Line2D at 0x1dc55d82c50>,
  <matplotlib.lines.Line2D at 0x1dc55d870d0>,
  <matplotlib.lines.Line2D at 0x1dc55d87c90>],
 'boxes': [<matplotlib.patches.PathPatch at 0x1dc55bc6290>,
  <matplotlib.patches.PathPatch at 0x1dc55d808d0>,
  <matplotlib.patches.PathPatch at 0x1dc55d61710>],
 'medians': [<matplotlib.lines.Line2D at 0x1dc55d6b3d0>,
  <matplotlib.lines.Line2D at 0x1dc55d83d10>,
  <matplotlib.lines.Line2D at 0x1dc55d88750>],
```

```
'fliers': [<matplotlib.lines.Line2D at 0x1dc55d3dd10>,
 <matplotlib.lines.Line2D at 0x1dc55d6ae50>,
 <matplotlib.lines.Line2D at 0x1dc55d89210>],
'means': []}
```



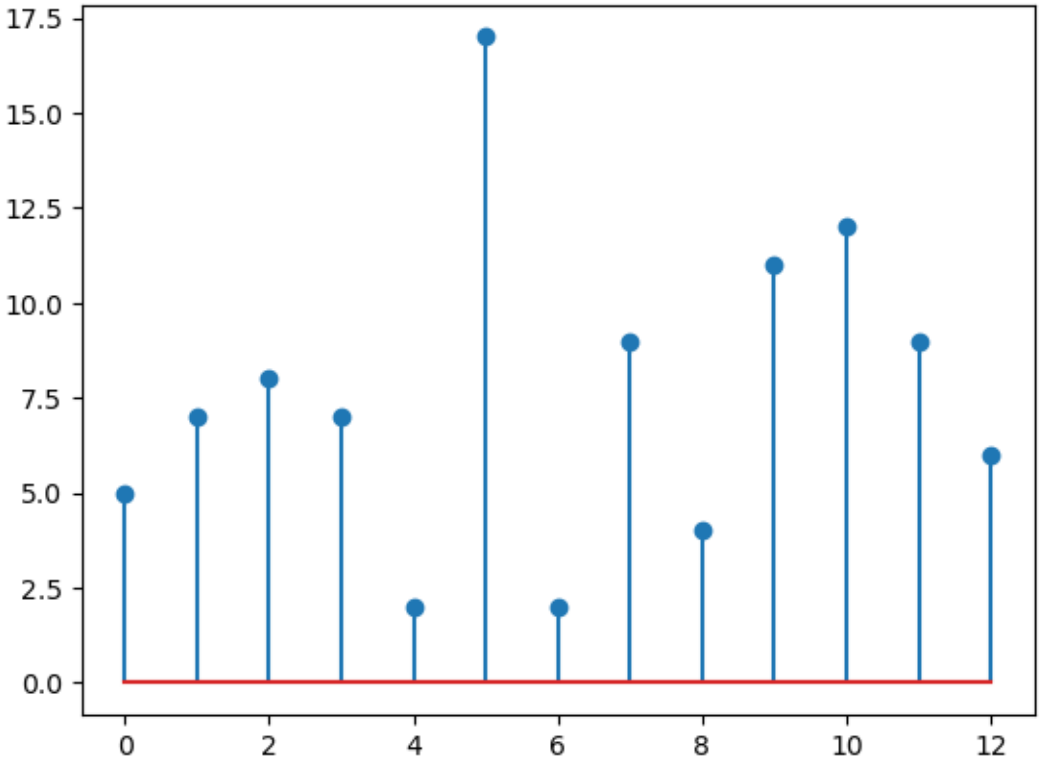
[35]:
```
data
```

[35]:
```
[array([ 0.08905337,  0.7788969 ,  1.26464491, -0.88051133,  0.2364056 ,
         0.81560447,  1.86081167,  0.25559049, -0.54150372, -0.68959966,
        -0.35744073, -0.6519202 ,  0.82653585,  1.06930572,  0.72485682,
         1.19218624, -0.45376854,  0.38033506, -0.38466318,  0.04365869,
         1.22498574, -0.02973531, -1.8648058 , -0.25281599, -0.7128498 ,
        -1.50891712, -0.79036569,  0.9606248 ,  1.68091065, -0.48900604,
         1.00253584,  1.1782221 , -1.15979227, -0.0393627 , -0.04446017,
         0.17238568, -1.59375081, -0.34914224,  1.05782121,  1.2622032 ,
         1.83136208, -0.33750905,  1.86950756,  0.66590511, -1.35920117,
         0.76160928, -0.35228003,  0.51907626, -0.10252394,  1.20823864,
         0.25656016, -0.28250502,  0.96496577,  0.25622178, -0.4129564 ,
         1.27727436, -0.40834524, -0.63713486, -0.53957461, -1.46547209,
        -0.55320717,  1.86087769, -0.90828394,  0.0084189 , -1.10818335,
        -0.61135315,  1.51869216,  0.89635574, -0.61025035,  0.00622012,
        -0.82600396, -0.78420535, -0.9148222 , -0.89713978,  0.32594927,
```

```
        0.59679317,  0.48824249, -0.16943686, -1.3580452 , -0.06711047,
       -0.92429374,  0.88130113,  0.55644294,  0.74689153, -0.34836832,
       -1.81101917,  0.95789639,  1.22632133, -1.48331661,  0.13791604,
        1.19325258, -1.07572355,  1.76768278, -0.34903246, -1.07539011,
        1.37165804,  0.29453214, -0.65313816,  0.87978795, -1.95576713]),
 array([ 3.63061275, -1.63250012,  0.57196855, -0.79723234,  0.87246423,
        1.77585145, -1.64777221,  1.57272584, -0.5880665 , -1.37571902,
       -0.44808721, -2.00921721, -1.59133515,  1.89826768, -0.30164608,
        2.2830662 ,  0.86351399, -2.05193659, -3.79574286, -1.37448364,
        0.52825298,  0.01762584, -1.66019124,  2.0256246 , -3.49144701,
       -2.51729317,  0.2545902 ,  4.38752631,  0.82786826, -2.01047932,
        1.18531428,  1.36679315,  1.87571386, -0.81972907, -1.59945288,
        0.23643456,  0.83054976,  3.1964705 ,  4.17929775, -1.46613987,
       -1.04594325, -0.12714616, -0.49954421,  0.97158035,  0.97147441,
        0.51579397,  2.68814258, -0.31702185, -2.10161588, -3.12120245,
       -2.05627975, -2.08851584, -3.76684766,  1.32637554,  0.37053567,
       -0.84558545, -1.77977796, -3.16189075, -1.83943919,  3.69581811,
       -4.42158328,  0.28362103,  1.48727394,  0.27226446, -0.63360984,
        1.53614789, -2.20665652,  1.11108999,  2.86139474, -2.08274877,
       -1.63966681,  0.27188075,  1.97914559,  2.67829659,  1.36125391,
       -1.33900555, -2.22882542,  0.48595715,  0.79651525, -2.20110462,
        0.11958097,  0.66447493,  2.09012799, -1.98112541,  1.11392906,
       -2.33855159, -0.85317196, -0.1565535 ,  3.03583252,  6.62860848,
        0.50709135,  0.99058299,  2.01508323, -2.72331245,  1.5313268 ,
       -2.48347972,  3.25822395,  2.88698353,  1.00307937, -1.61339703]),
 array([ 5.00407581,  0.77906512,  0.33073063,  1.73863004, -5.13703971,
        1.28549827,  2.98832446,  1.01700294,  0.43577624, -2.86097435,
        1.93066449,  0.60140684,  0.02719136, -1.68961476,  2.56851333,
        3.74155311,  1.16443852,  2.69024722,  3.97806989,  4.76448344,
        0.67695045, -0.63998406, -1.24483761,  0.04124104,  1.65589922,
        2.81454581, -1.28022606,  0.44957267, -2.98974005,  3.79189202,
       -1.17438316,  1.07376338, -2.15811019, -0.89482812, -2.05798831,
        5.11489967,  1.07809123, -0.499334  , -2.84292695, -1.83089914,
        1.70611519,  4.99777505, -4.29266649, -2.75113968, -0.52736449,
       -1.88480205, -4.97754621,  0.05792925,  2.25312765, -4.73133408,
        0.63287933,  2.89401873,  2.94098198, -1.25961208,  6.91424714,
        1.66534021, -2.43319865,  2.95289682, -3.36776391,  2.68580836,
       -3.78616303,  2.37162772, -1.7506853 ,  3.39686419,  2.22203224,
        1.34162376,  0.1758973 ,  3.26104877, -5.41002574,  1.29766301,
        1.2402981 ,  1.68644968,  2.47868647, 11.10737471,  0.350348  ,
        2.4134986 ,  0.45710549,  4.7180042 , -6.06243145,  5.09384259,
       -2.05141925, -6.78672388, -3.23455673, -0.52355801,  0.58273575,
       -0.13767409,  2.45045317, -0.10553096, -0.1895928 , -6.43054658,
        1.4620322 ,  3.0307251 , -1.26201849, -5.6312529 , -4.00951647,
        5.02068491, -1.81862174, -1.99375165, -2.45835328,  2.04265235])]
```

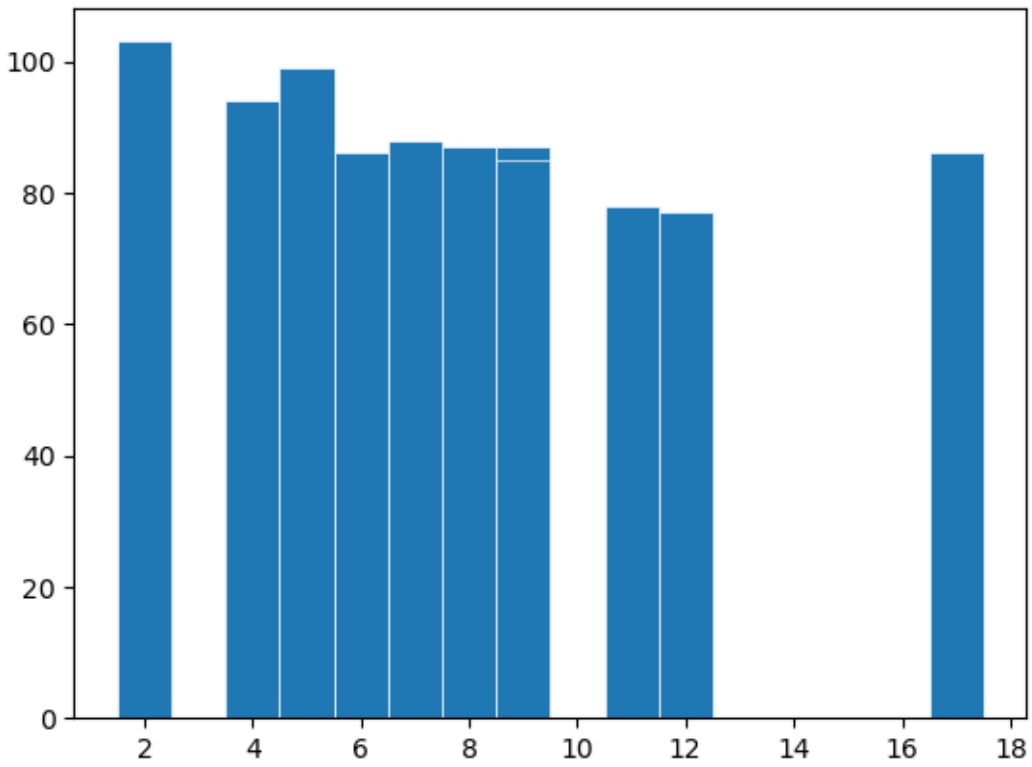
[36]:
```
plt.stem(x)
```

[36]: <StemContainer object of 3 artists>



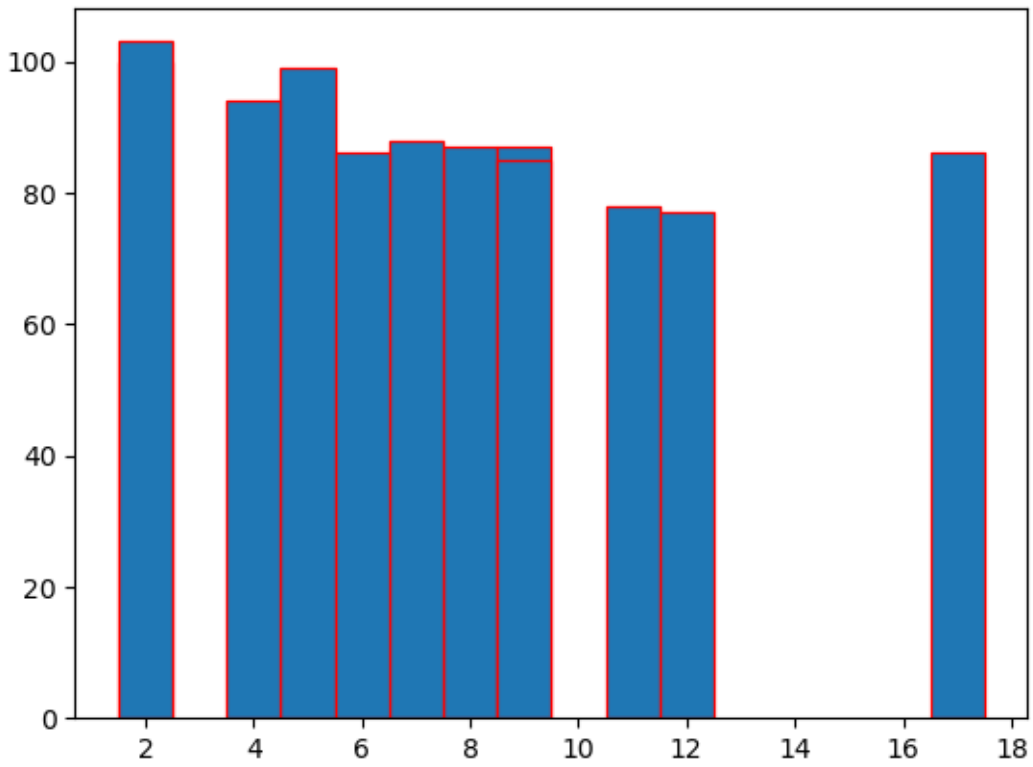
[37]:
```
plt.bar(x,y,width=1,edgecolor='white',linewidth=0.5)
```

[37]: <BarContainer object of 13 artists>



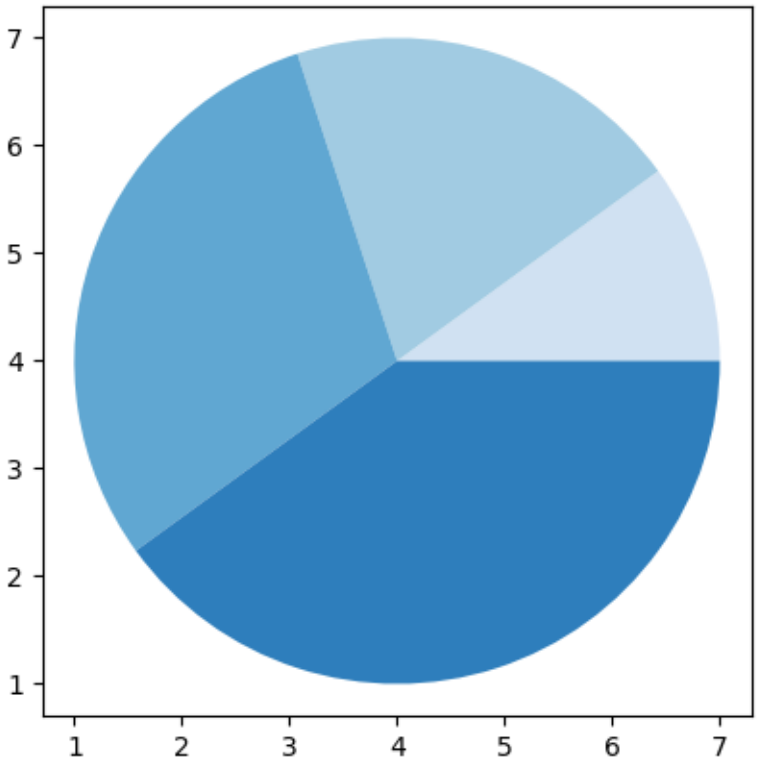
[38]:
```
plt.bar(x,y,width=1,edgecolor='red',linewidth=0.9)
```

[38]: <BarContainer object of 13 artists>



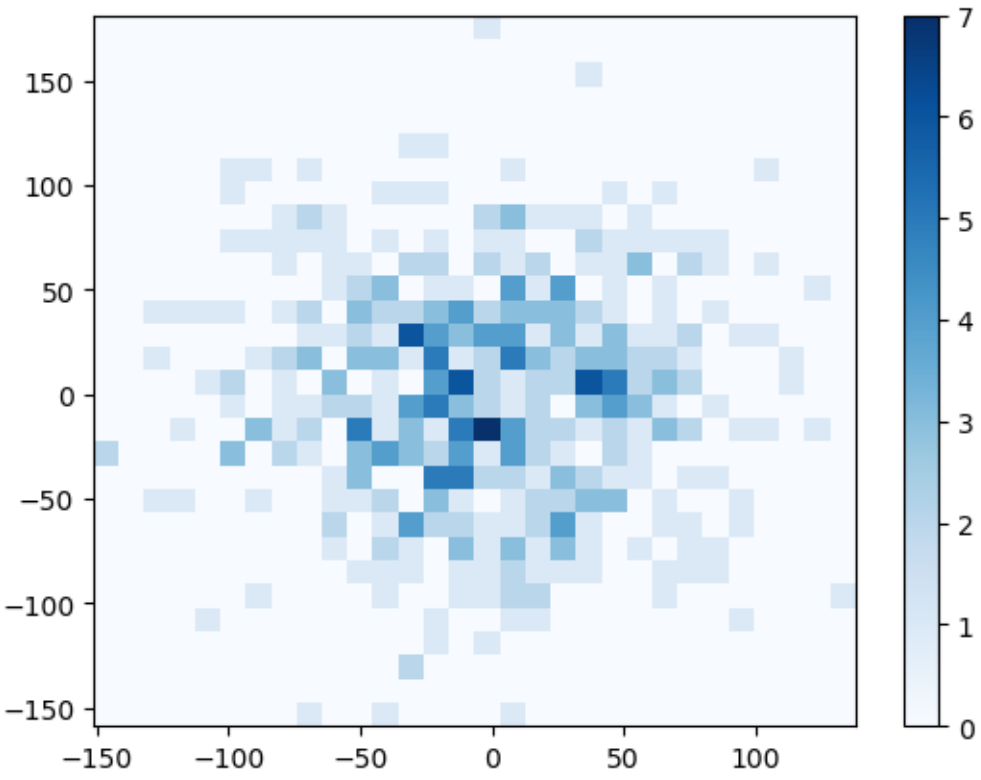
[39]:
```
x=[1,2,3,4]
colors=plt.get_cmap('Blues')(np.linspace(0.2,0.7,len(x)))
plt.pie(x,colors=colors,radius=3,center=(4,4),frame=True)
colors
```

[39]:
```
array([[0.81411765, 0.88392157, 0.94980392, 1.        ],
       [0.63252595, 0.79764706, 0.88687428, 1.        ],
       [0.37673203, 0.6530719 , 0.82248366, 1.        ],
       [0.17914648, 0.49287197, 0.73542484, 1.        ]])
```



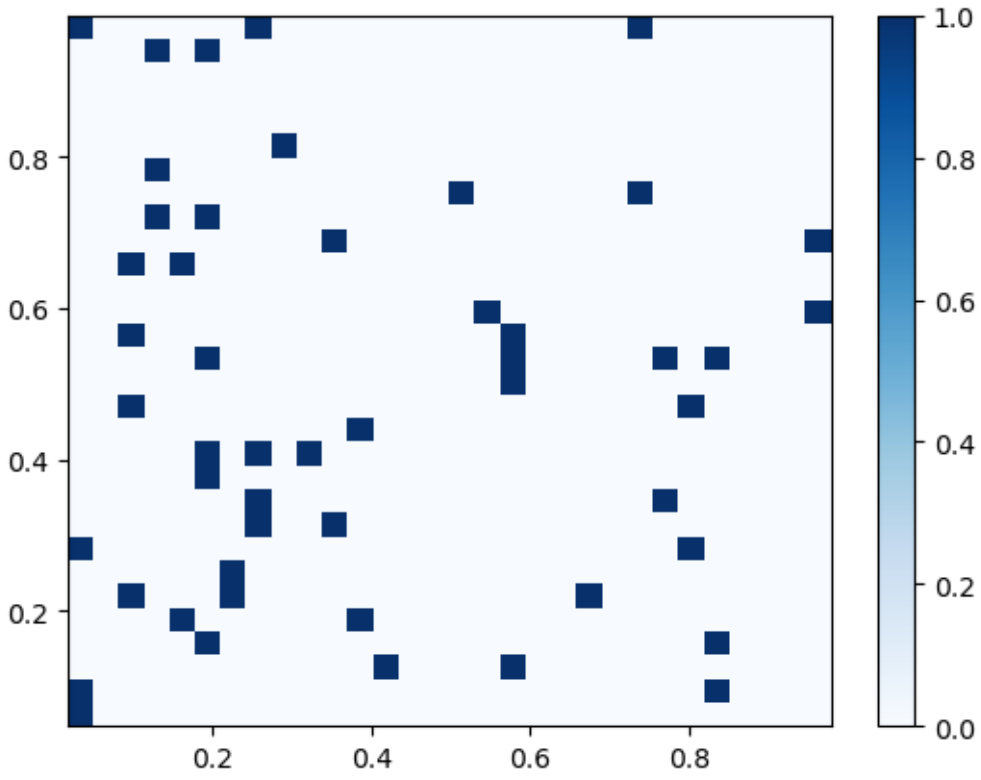
[40]:
```
x1=np.random.normal(0,50,500)
y1=np.random.normal(0,50,500)
plt.hist2d(x1,y1,bins=30,cmap='Blues')
plt.colorbar()
```

[40]: <matplotlib.colorbar.Colorbar at 0x1dc55fd0710>



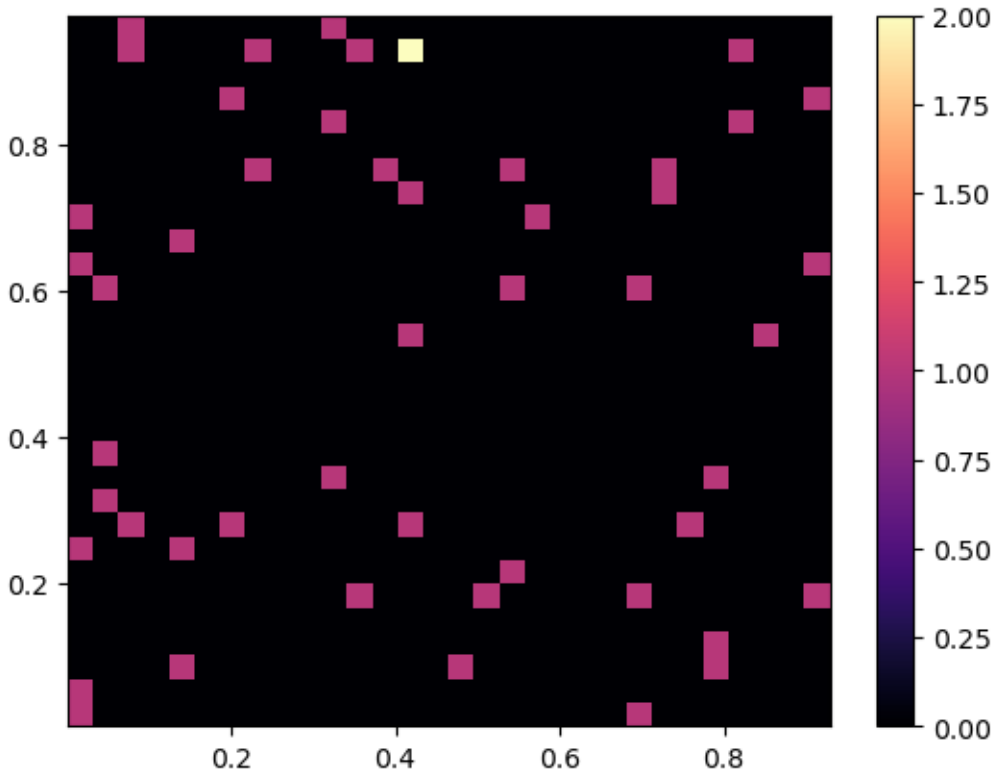
[41]:
```
x1=np.random.rand(50)
y1=np.random.rand(50)
plt.hist2d(x1,y1,bins=30,cmap='Blues')
plt.colorbar()
```

[41]: <matplotlib.colorbar.Colorbar at 0x1dc56078710>



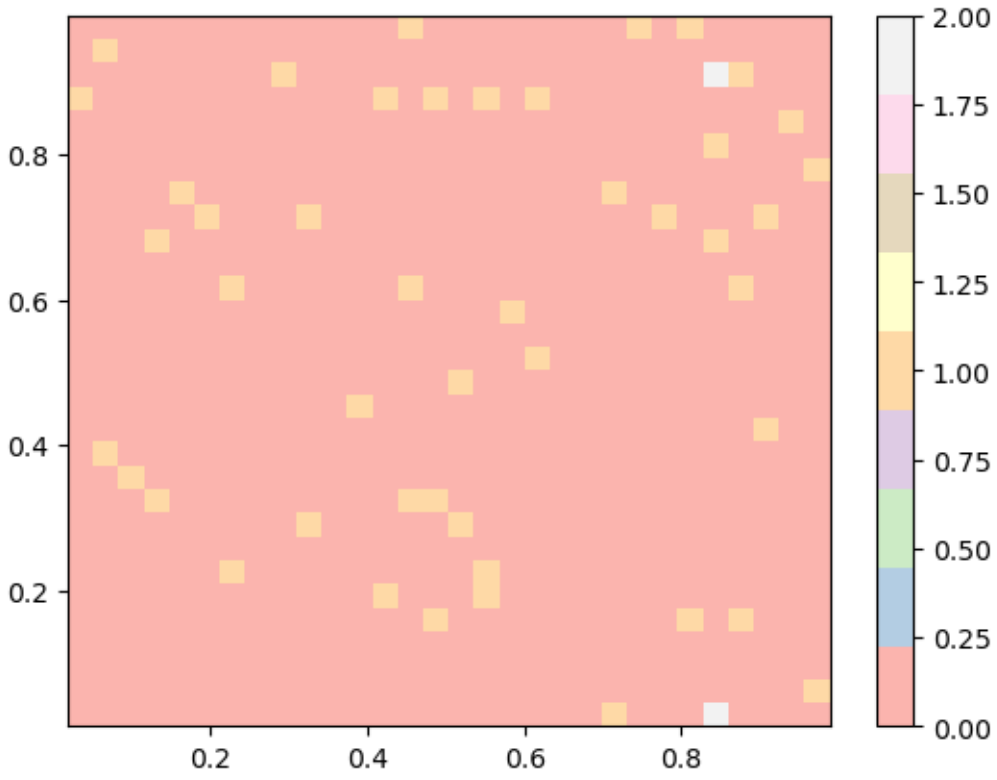
[42]:
```
x1=np.random.rand(50)
y1=np.random.rand(50)
plt.hist2d(x1,y1,bins=30,cmap='magma')
plt.colorbar()
```

[42]: <matplotlib.colorbar.Colorbar at 0x1dc560f3ed0>

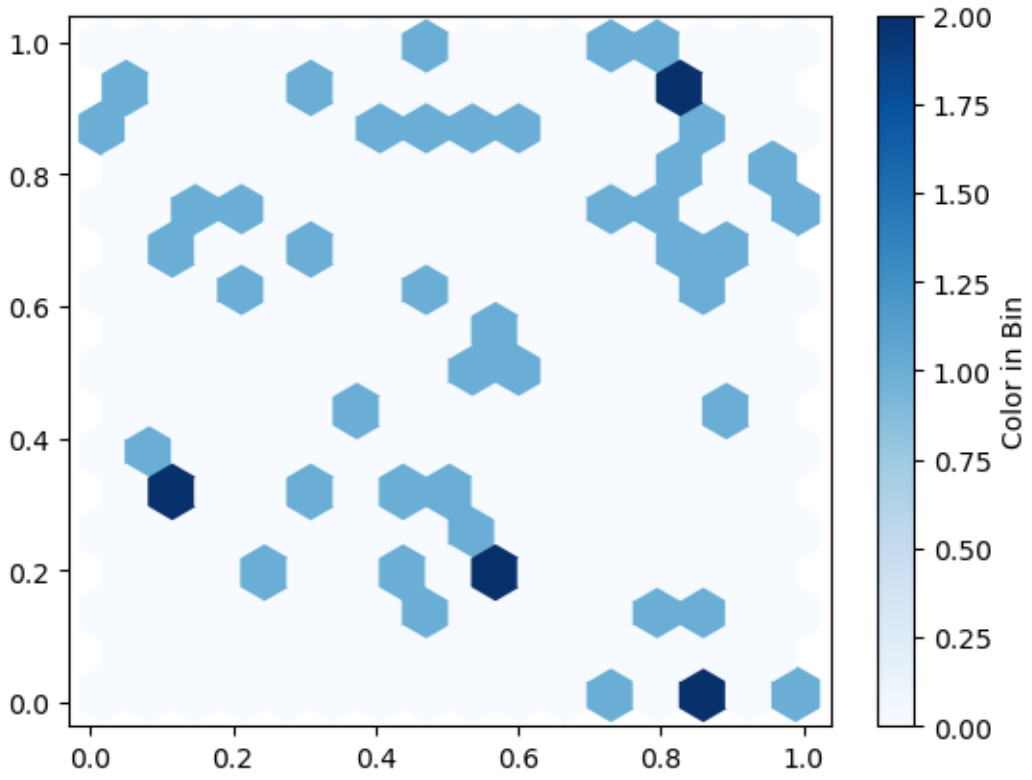


[43]:
```
x1=np.random.rand(50)
y1=np.random.rand(50)
plt.hist2d(x1,y1,bins=30,cmap='Pastel1')
plt.colorbar()
```

[43]: <matplotlib.colorbar.Colorbar at 0x1dc55a74510>

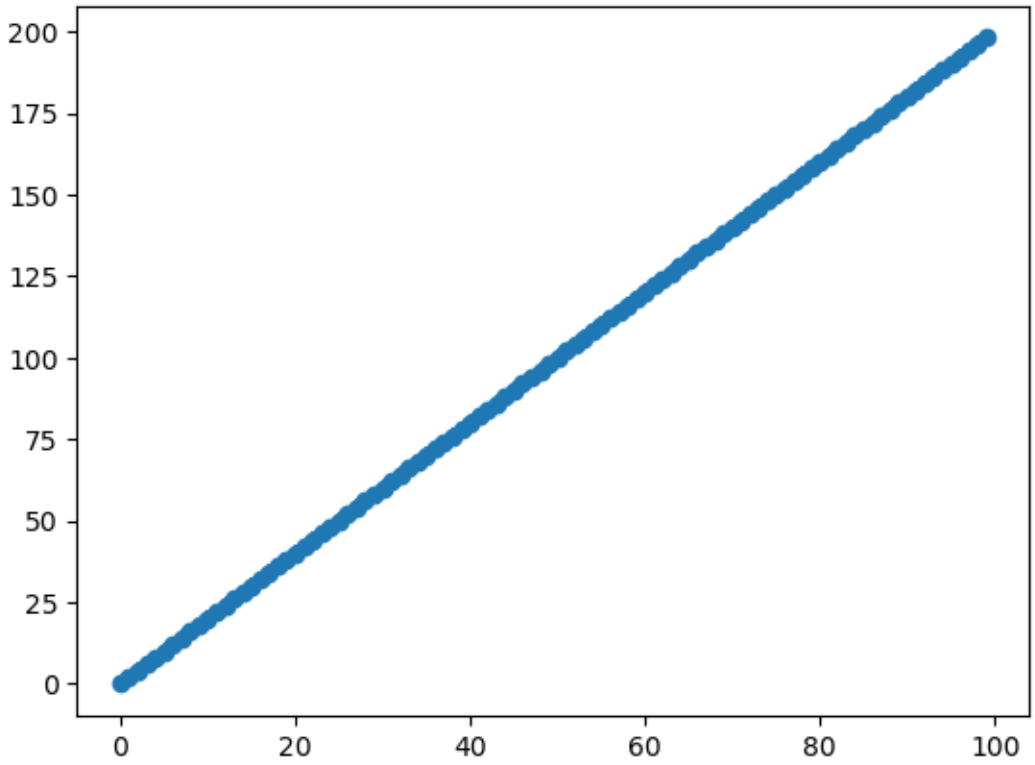


[44]:
```
plt.hexbin(x1,y1,gridsize=15,cmap='Blues')
cb=plt.colorbar(label='Color in Bin')
```



[87]:
```
plt.plot(x, y,drawstyle='steps',linestyle='-',marker='o')
```

[87]:
```
[<matplotlib.lines.Line2D at 0x1dc66ded390>]
```



[47]:
```
import seaborn as sns
%matplotlib inline
```

[48]:
```
dataset = sns.load_dataset('iris')
dataset.head()
```

[48]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

[88]:
```
tips = sns.load_dataset('tips')
tips.head()
```

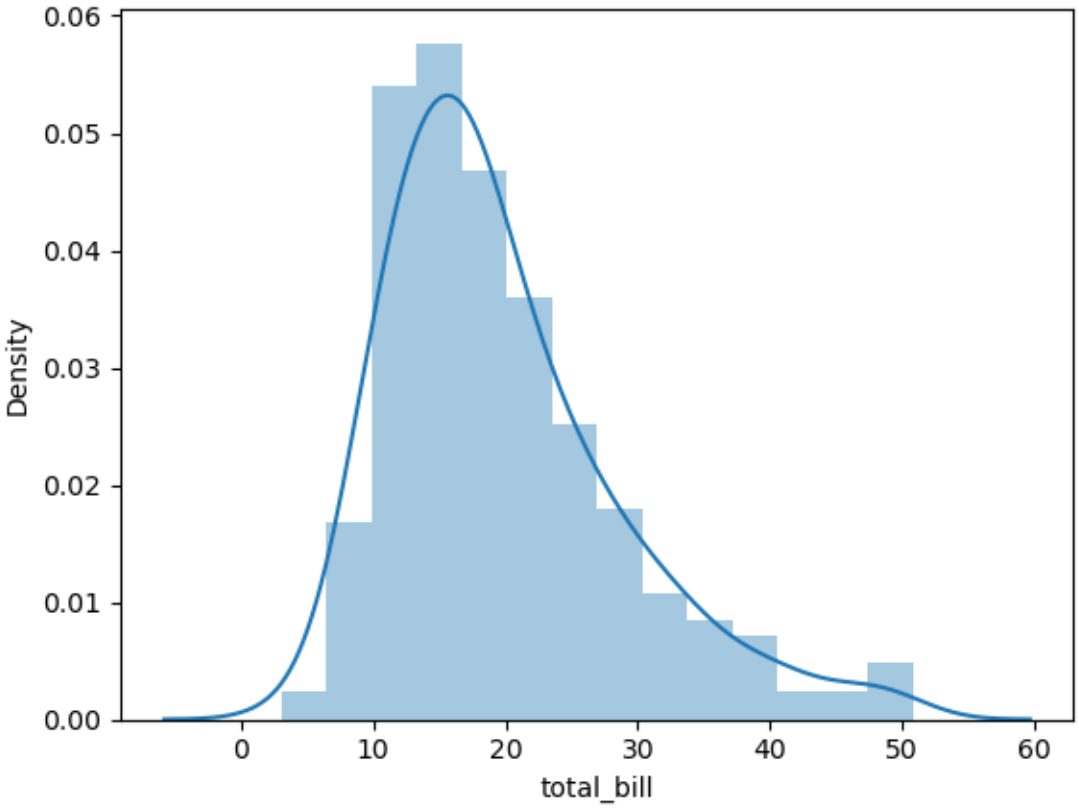
[88]:

	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

[56]:
```
sns.distplot(tips['total_bill'])
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\4271412032.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(tips['total_bill'])
```

[56]: <Axes: xlabel='total_bill', ylabel='Density'>

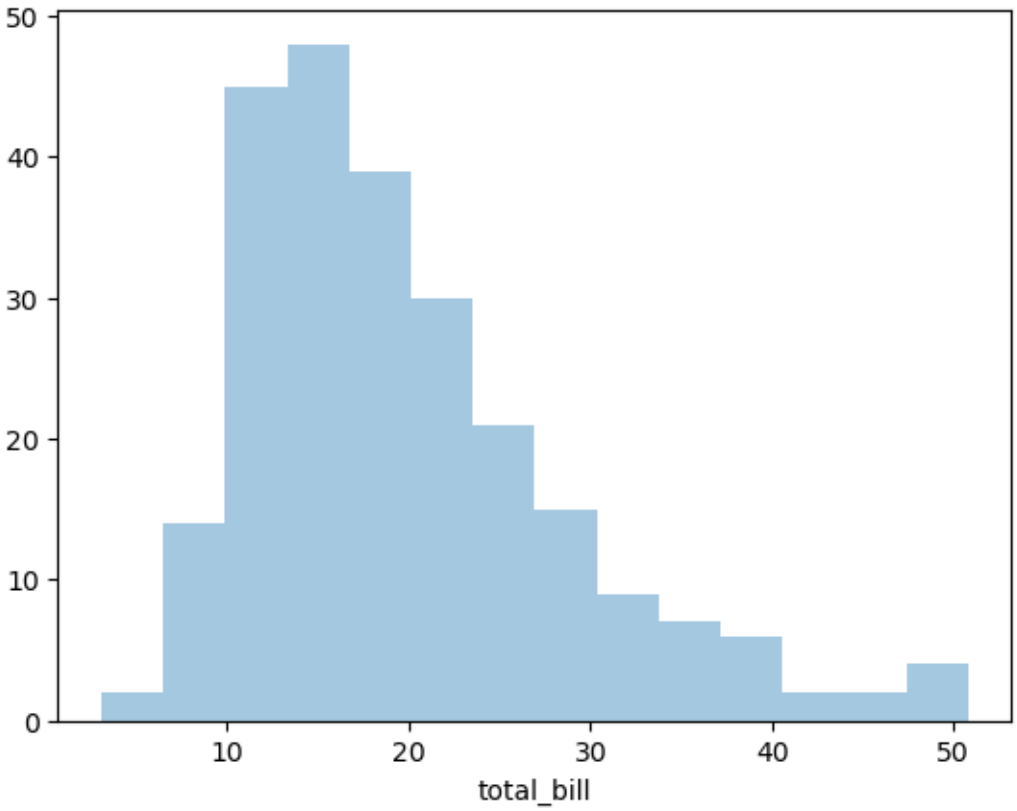


[57]:
```
sns.distplot(tips['total_bill'], kde=False)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\2337428669.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(tips['total_bill'], kde=False)
```

[57]: <Axes: xlabel='total_bill'>

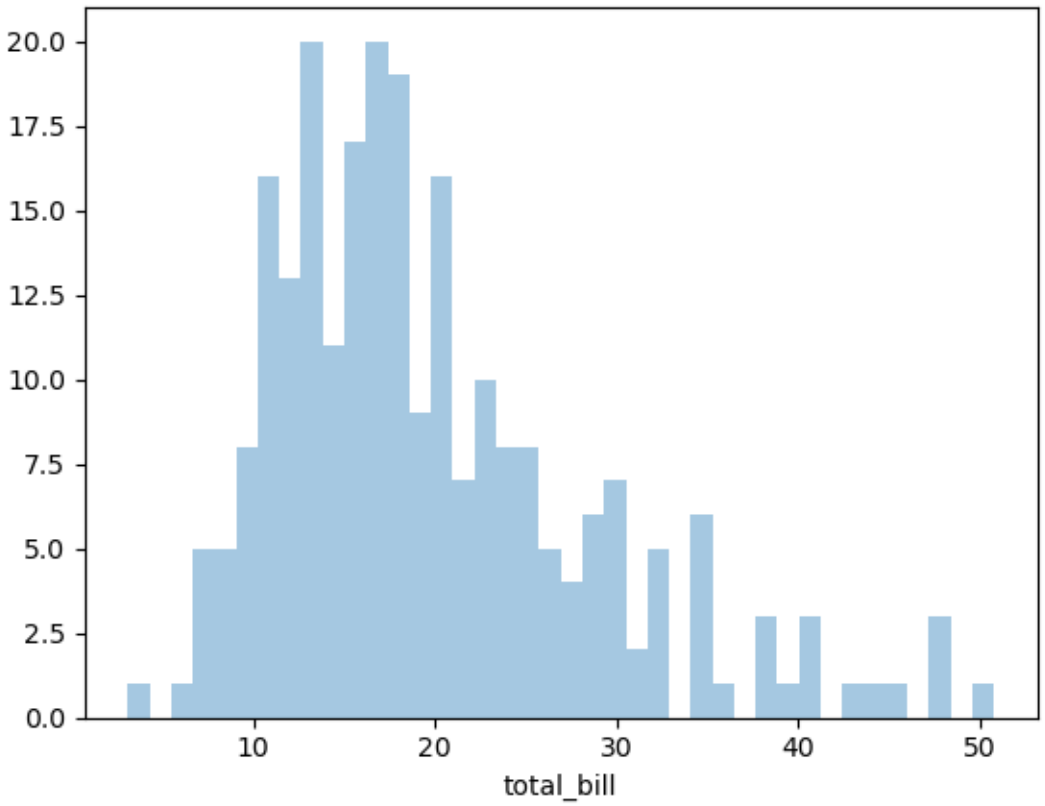


[59]:
```
sns.distplot(tips['total_bill'], kde=False, bins=40)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\1261316105.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sns.distplot(tips['total_bill'], kde=False, bins=40)
```

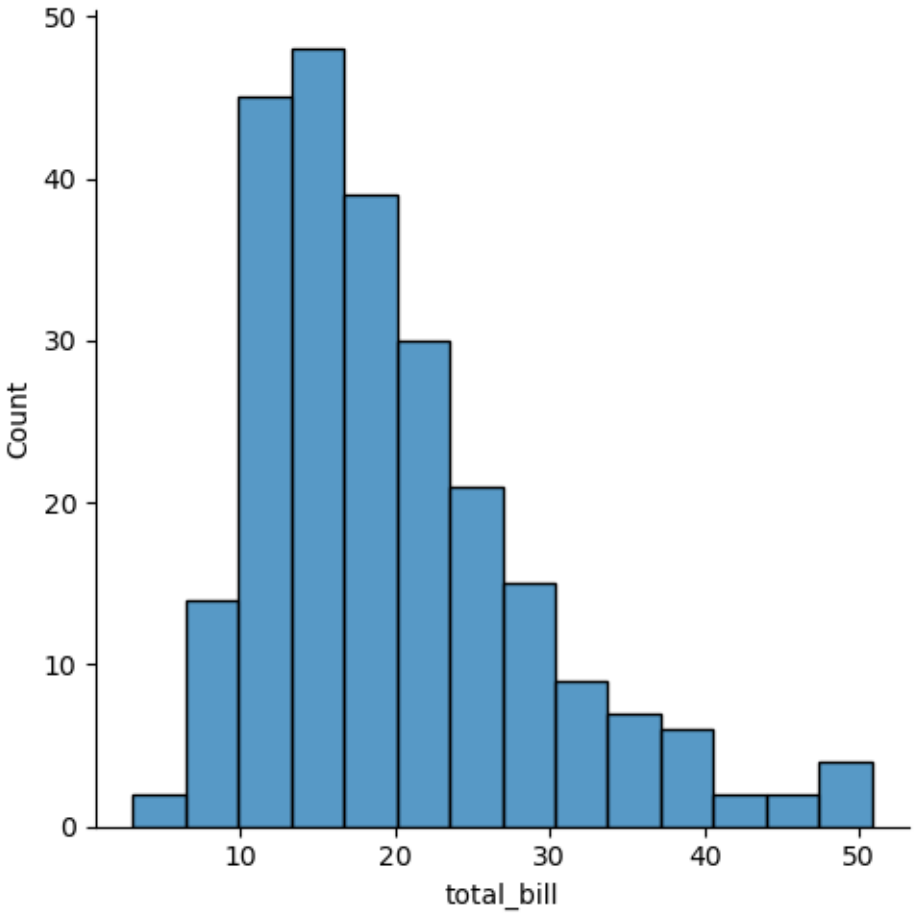
[59]: <Axes: xlabel='total_bill'>



[60]:
```
sns.displot(tips['total_bill'])
```

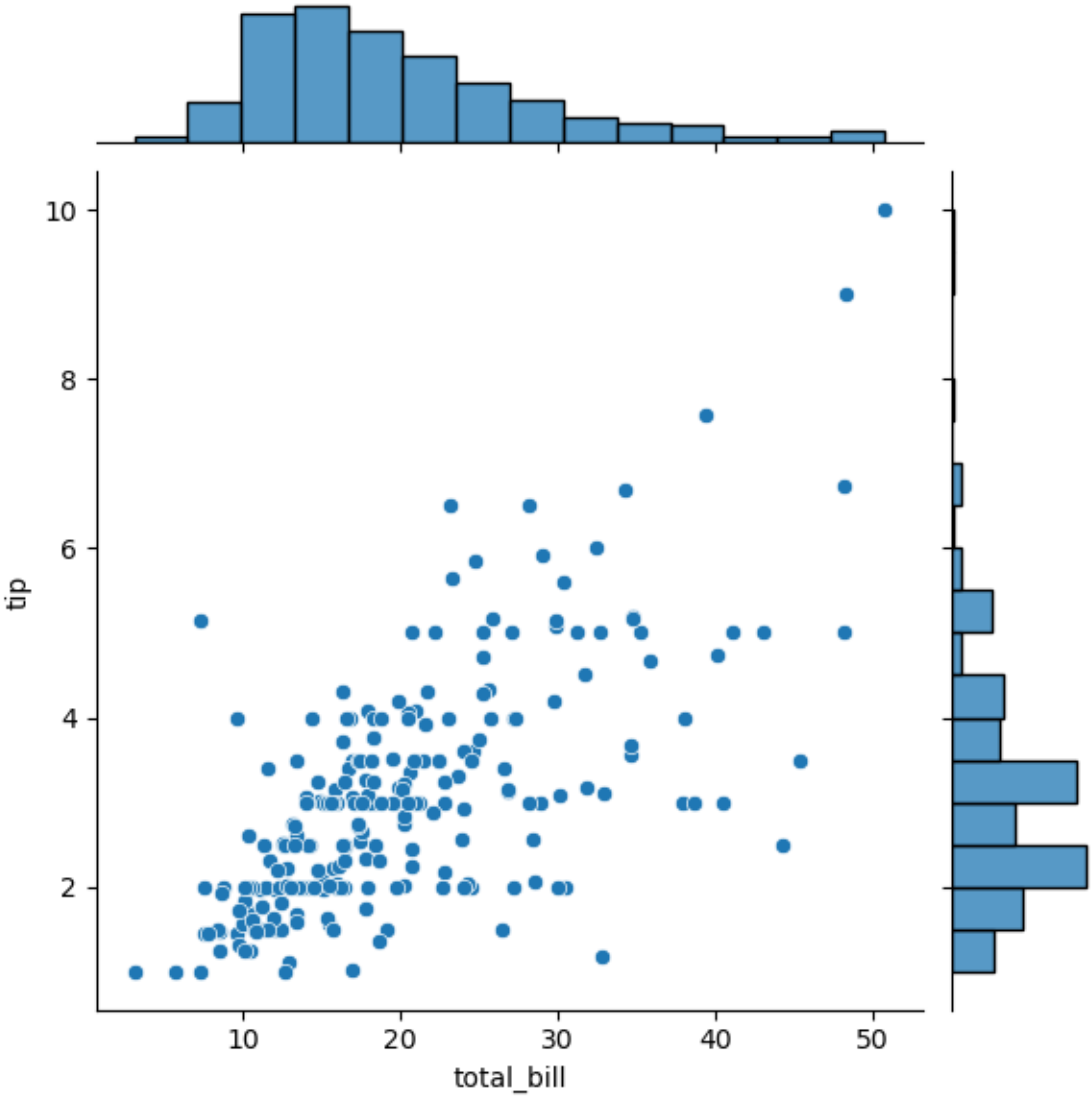
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[60]: <seaborn.axisgrid.FacetGrid at 0x1dc5e66ffd0>



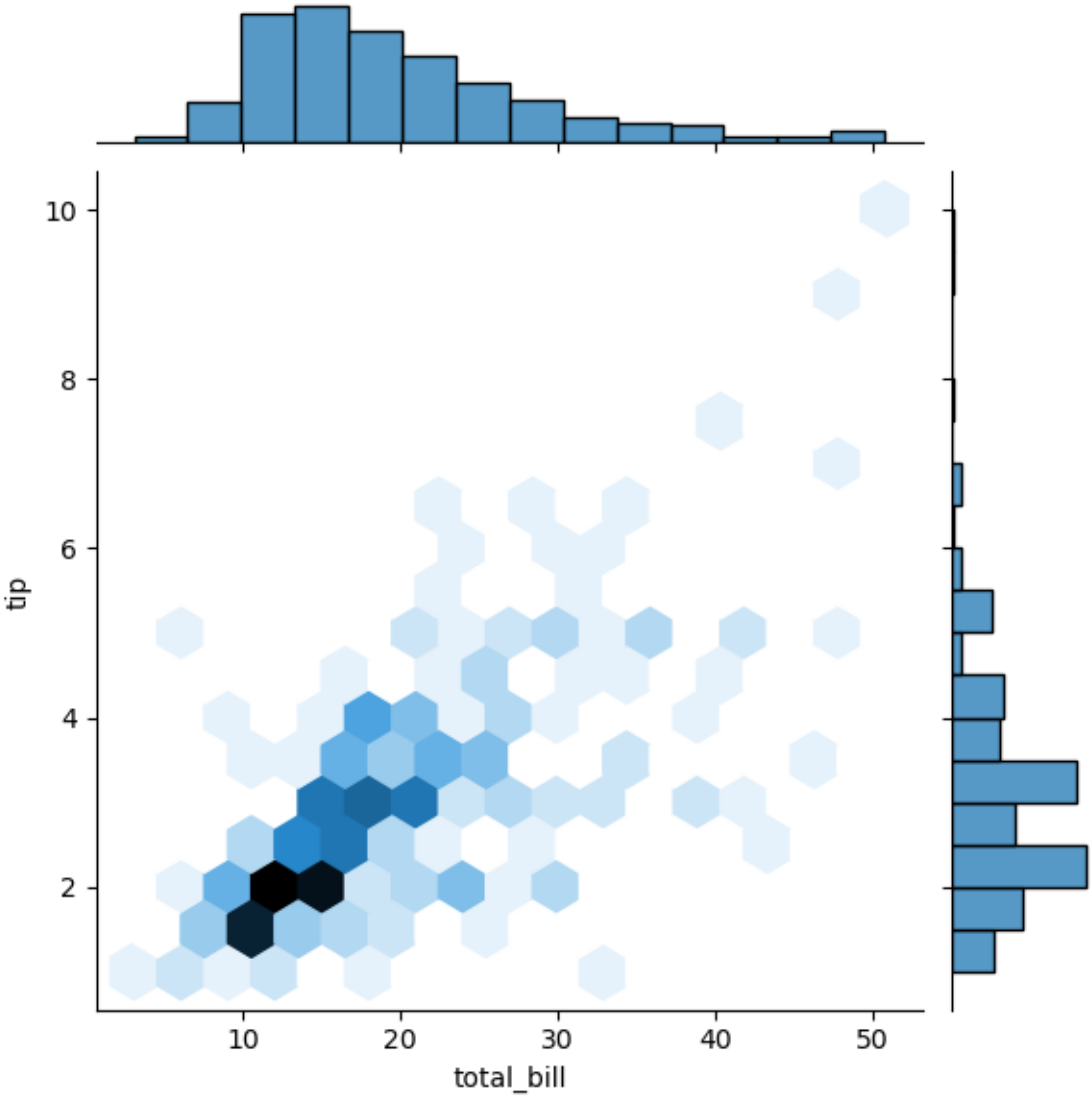
[62]:
```
sns.jointplot(x='total_bill', y='tip', data=tips)
```

[62]:
```
<seaborn.axisgrid.JointGrid at 0x1dc5e9545d0>
```



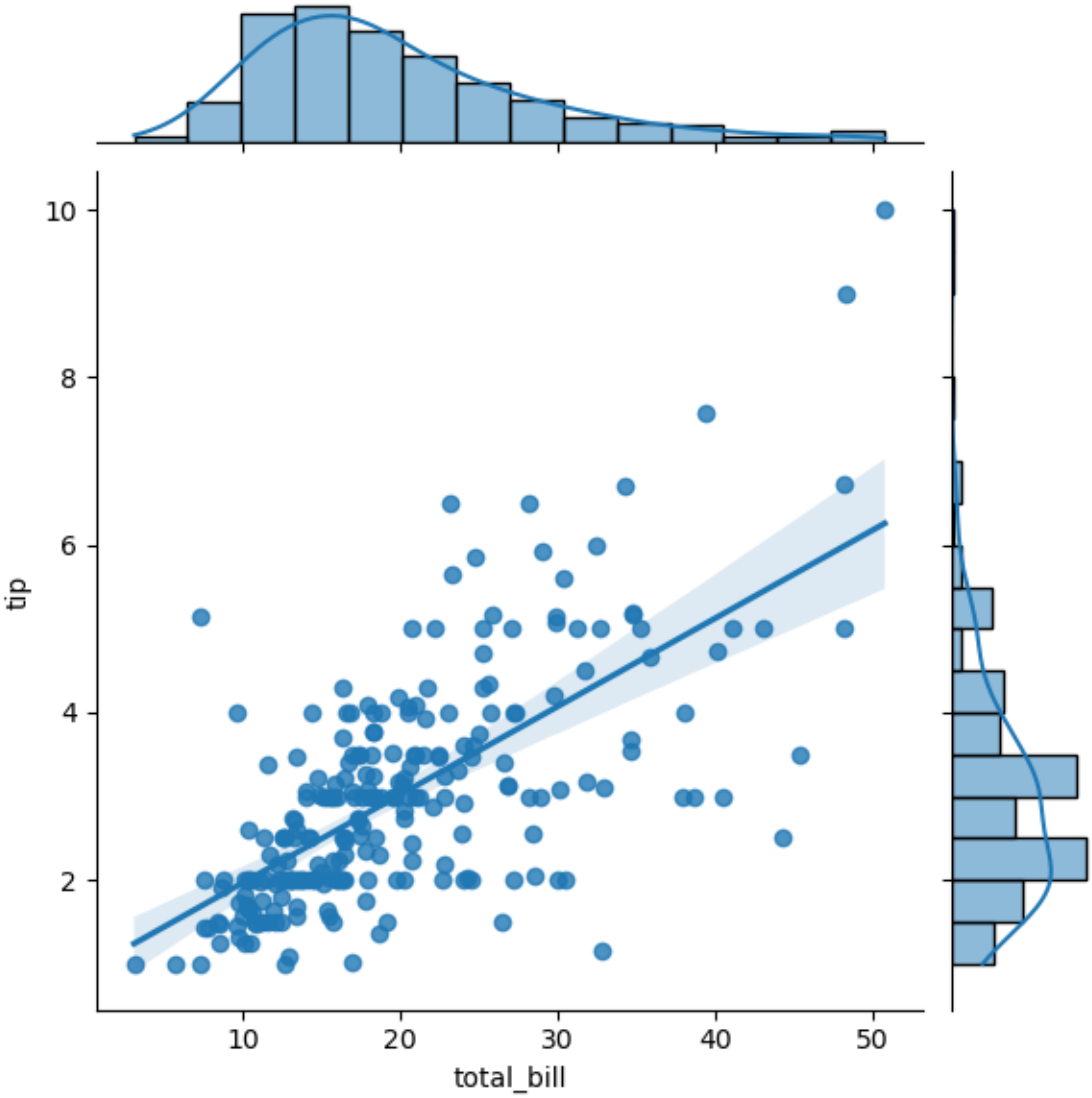
[64]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='hex')
```

[64]: <seaborn.axisgrid.JointGrid at 0x1dc5fe28b90>



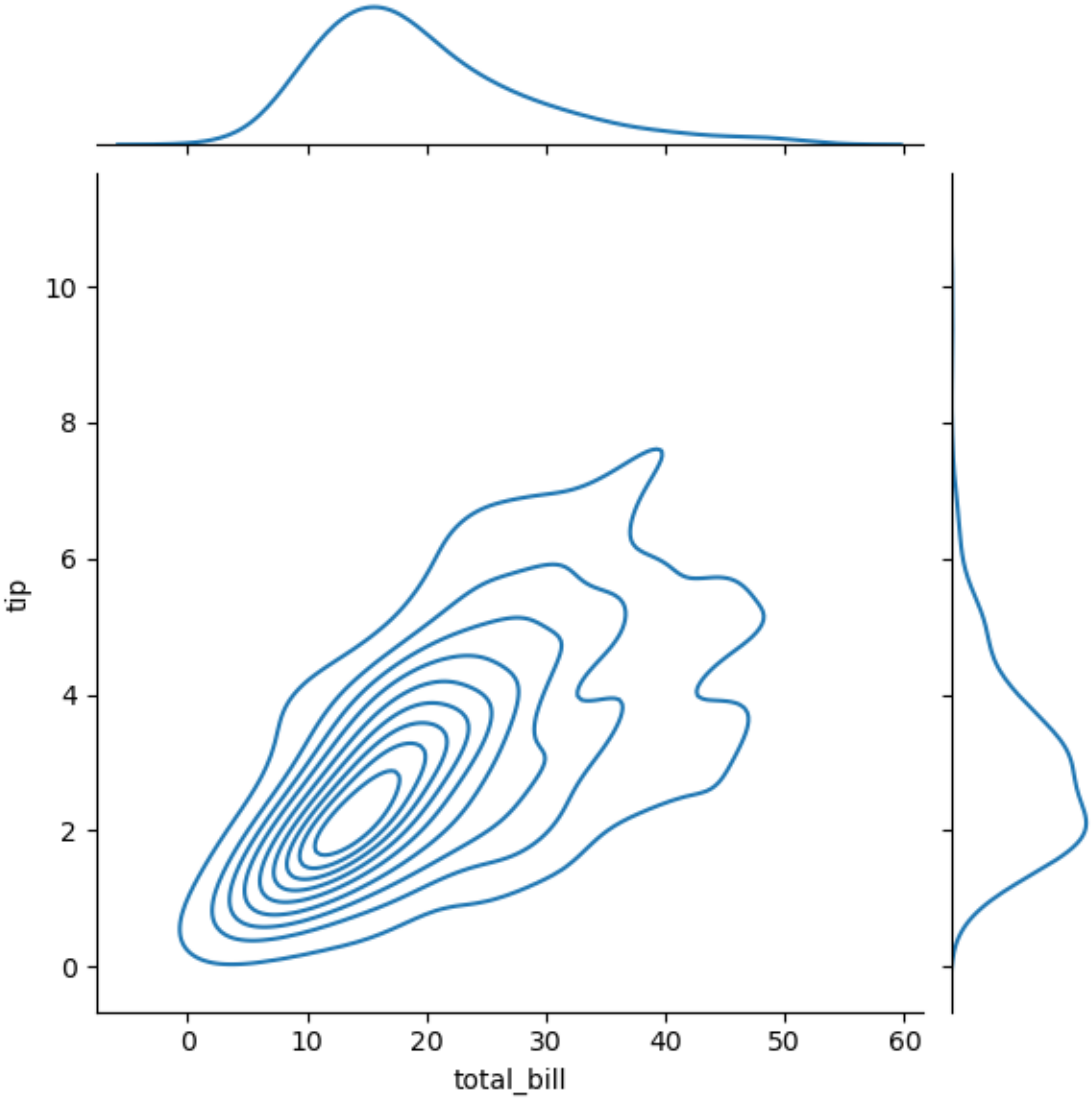
[65]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='reg')
```

[65]:
```
<seaborn.axisgrid.JointGrid at 0x1dc5fc1b410>
```



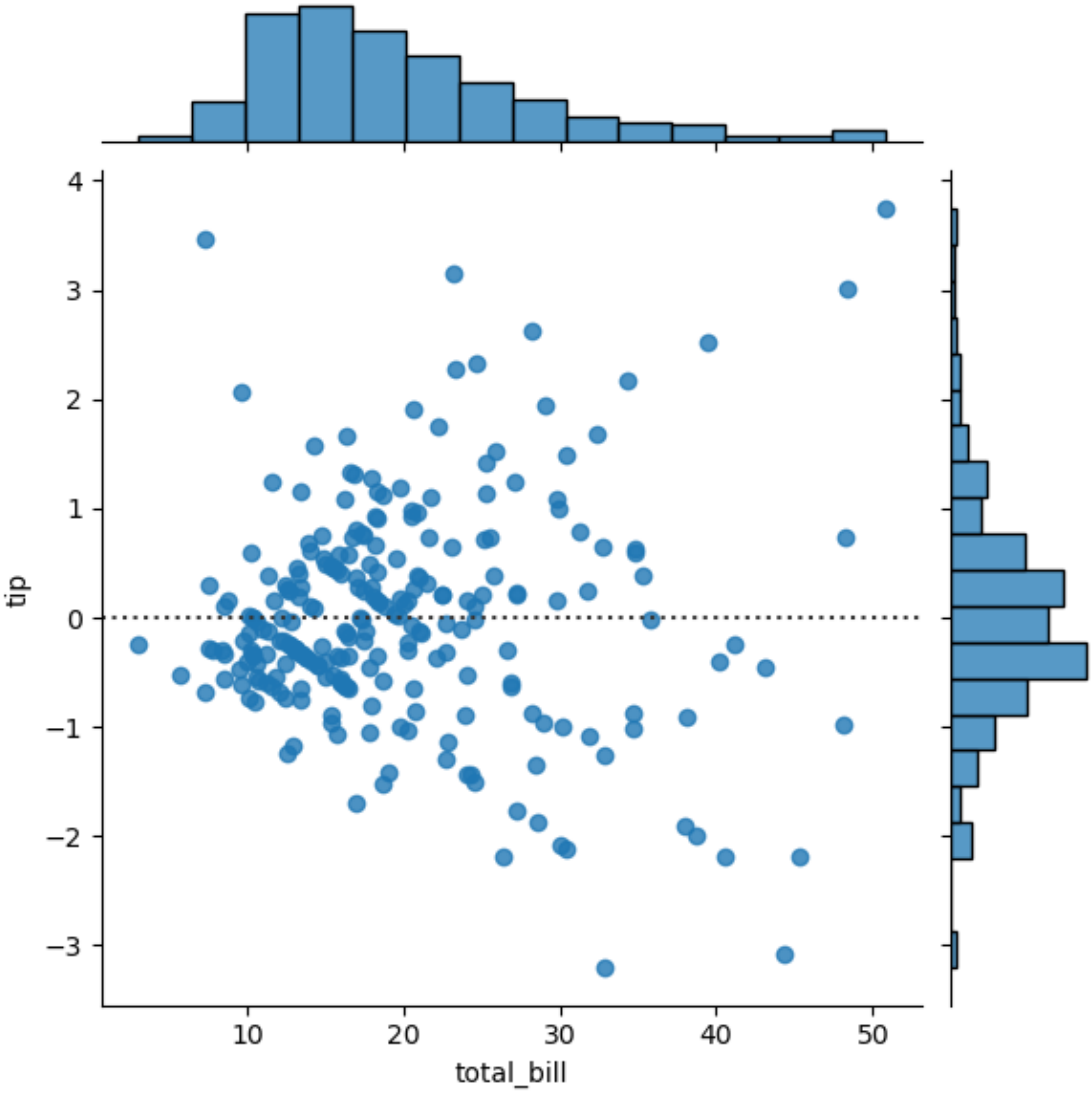
[66]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='kde')
```

[66]:
```
<seaborn.axisgrid.JointGrid at 0x1dc5ff46950>
```



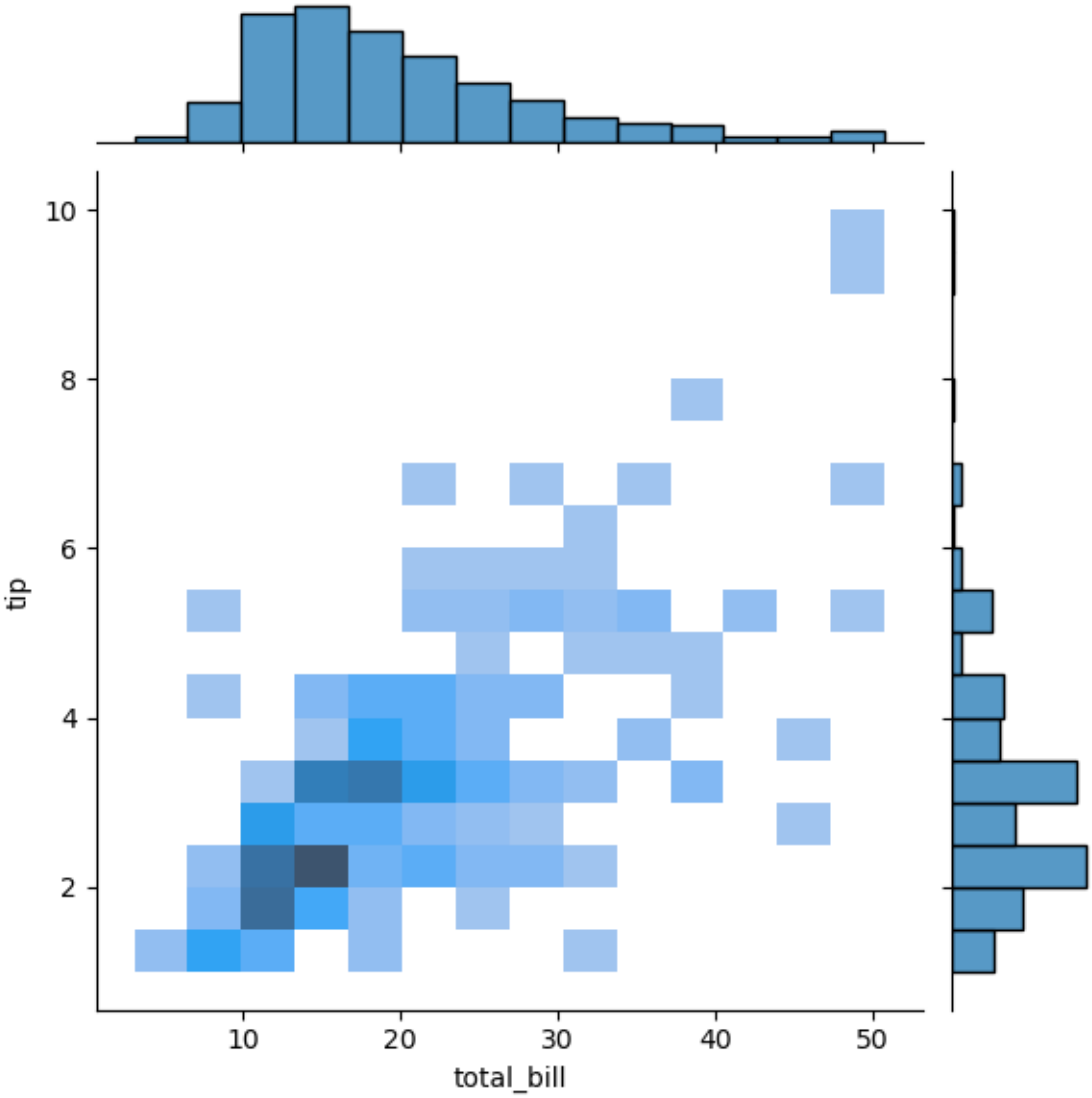
[67]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='resid')
```

[67]: <seaborn.axisgrid.JointGrid at 0x1dc607a5290>



[68]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='hist')
```

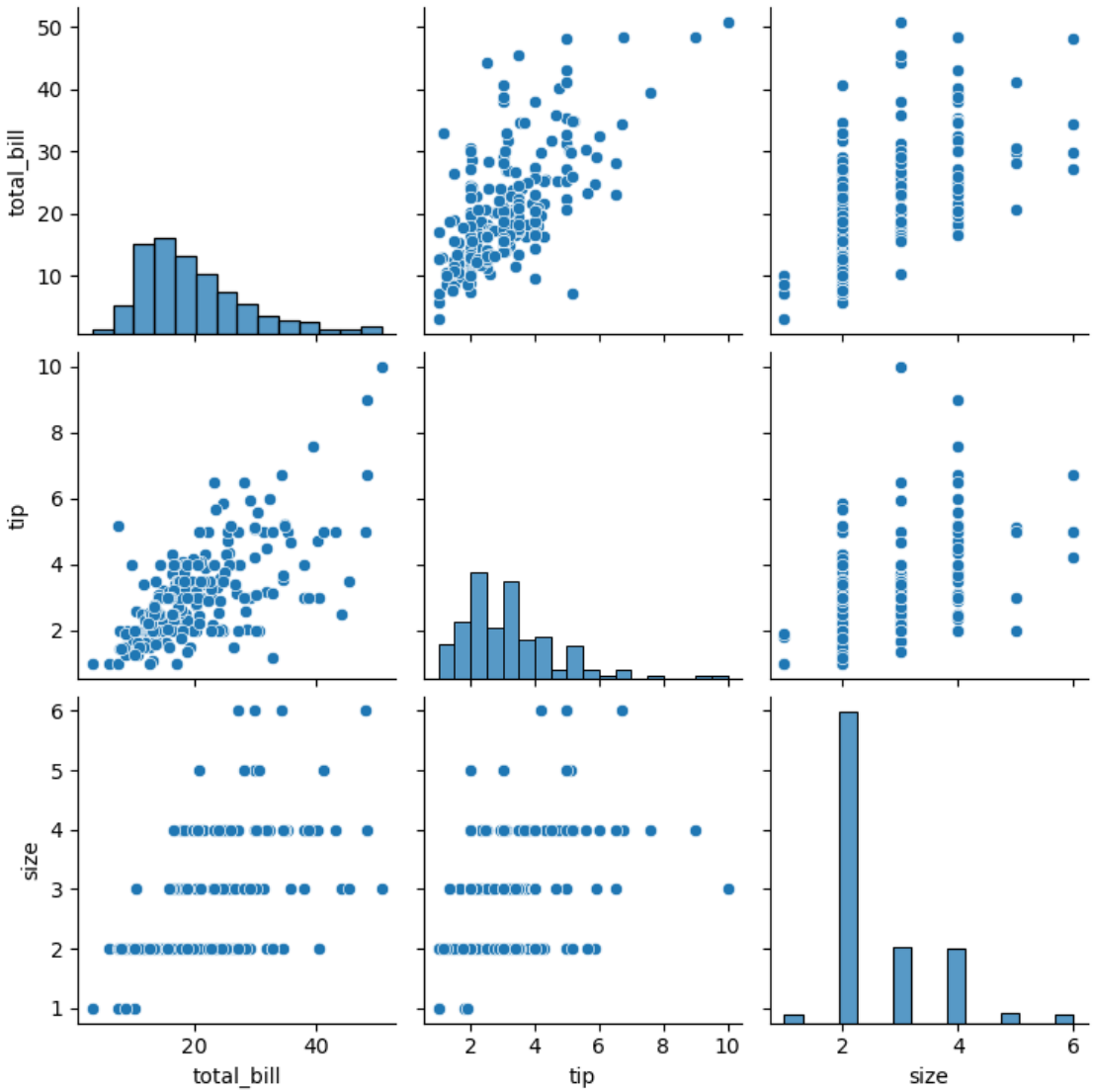
[68]:
```
<seaborn.axisgrid.JointGrid at 0x1dc6121e690>
```



[69]:
```
sns.pairplot(tips)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

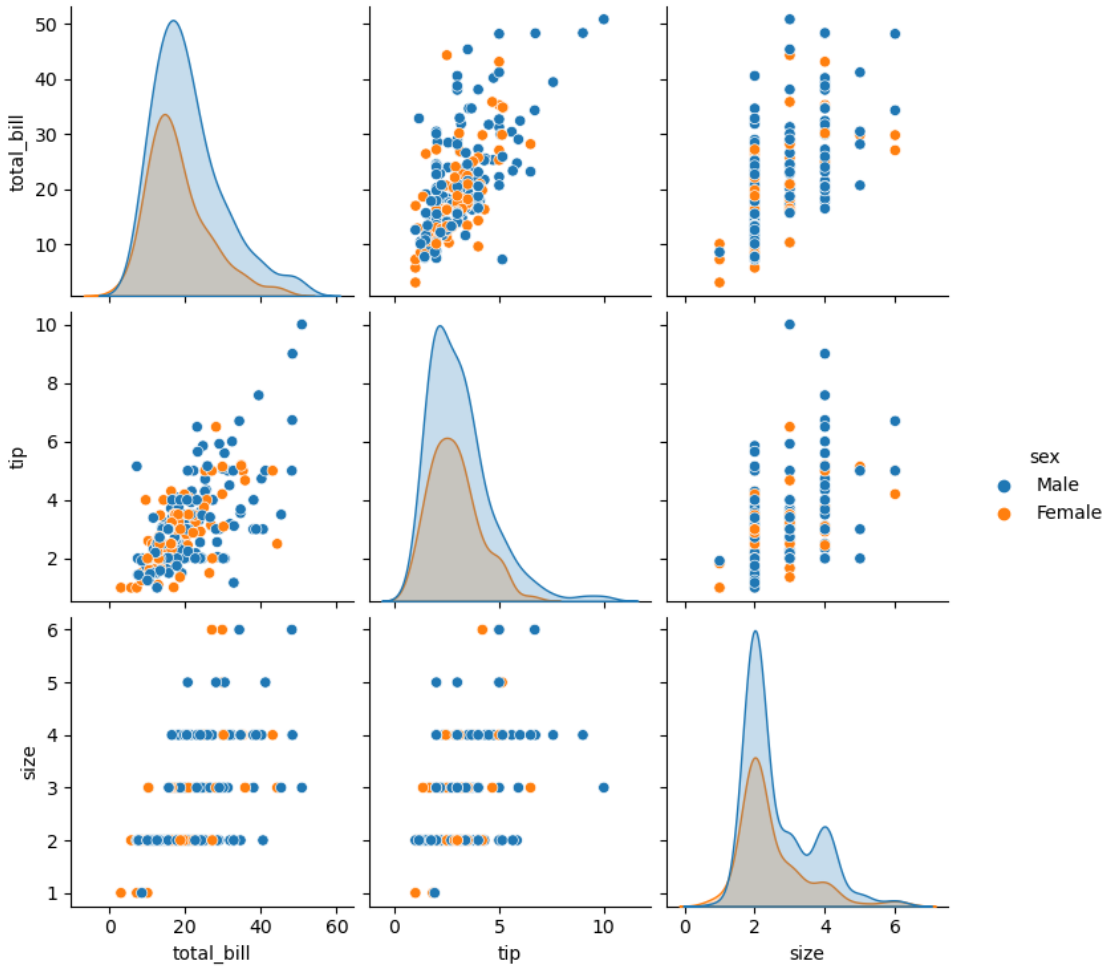
[69]:
```
<seaborn.axisgrid.PairGrid at 0x1dc6181b410>
```



[70]:
```
sns.pairplot(tips, hue='sex')
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

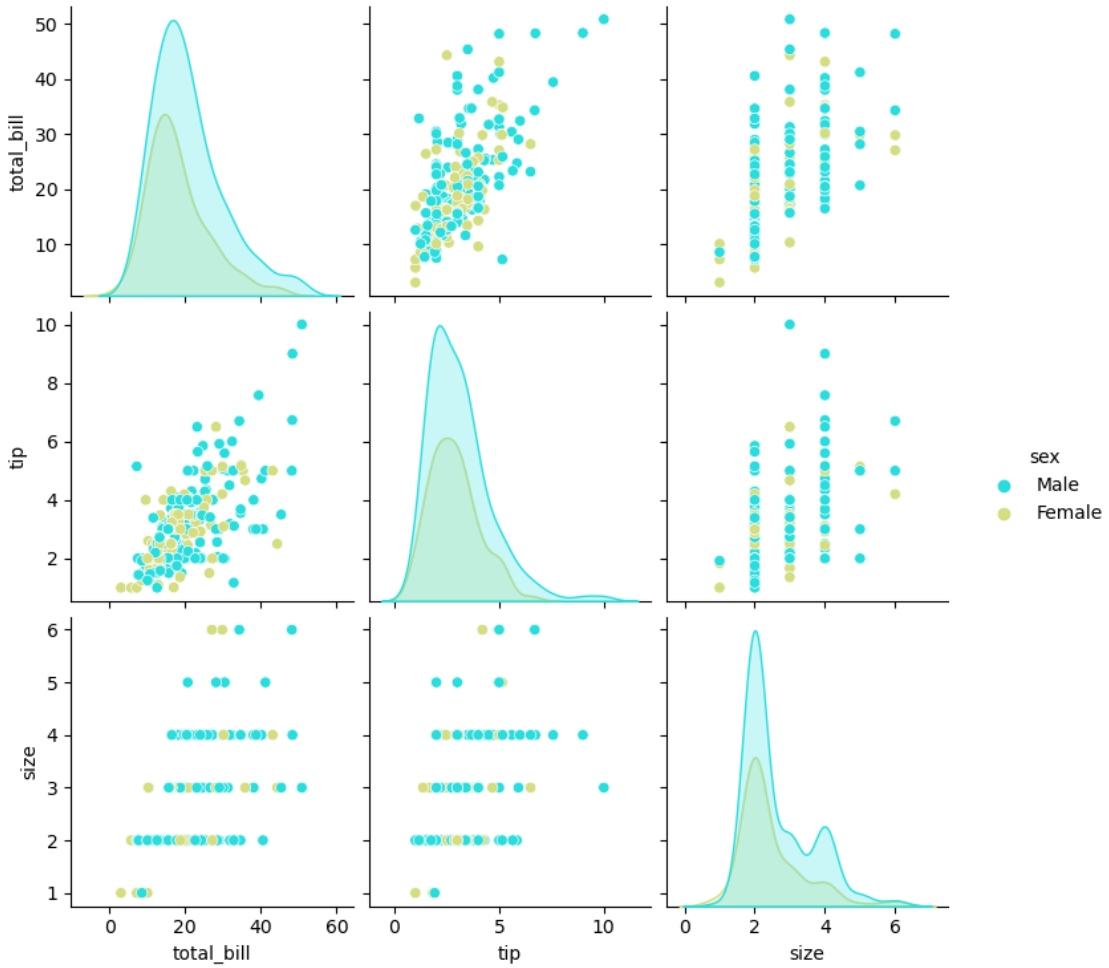
[70]:
```
<seaborn.axisgrid.PairGrid at 0x1dc62173ad0>
```



[71]:
```
sns.pairplot(tips, hue='sex', palette='rainbow')
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

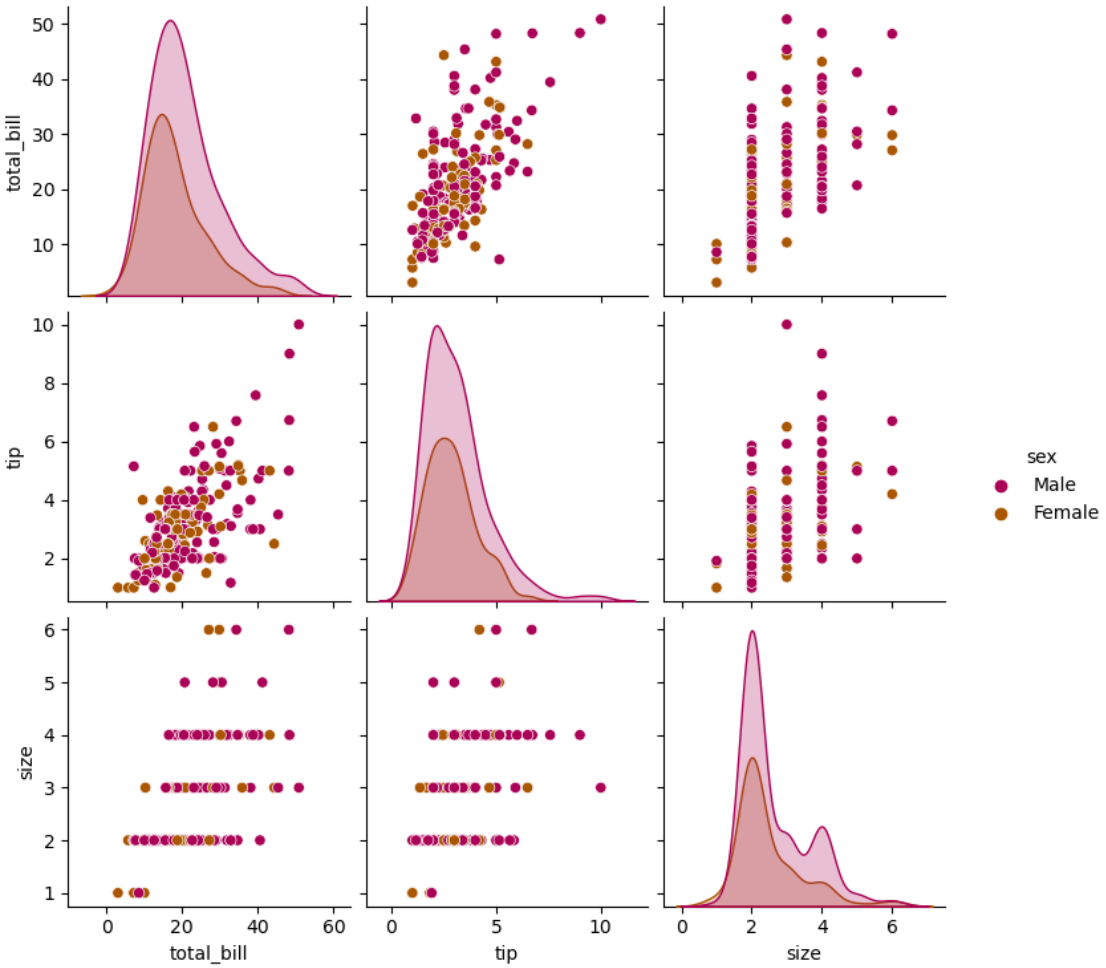
[71]:
```
<seaborn.axisgrid.PairGrid at 0x1dc621d6390>
```



[97]:
```
sns.pairplot(tips, hue='sex', palette='brg')
```

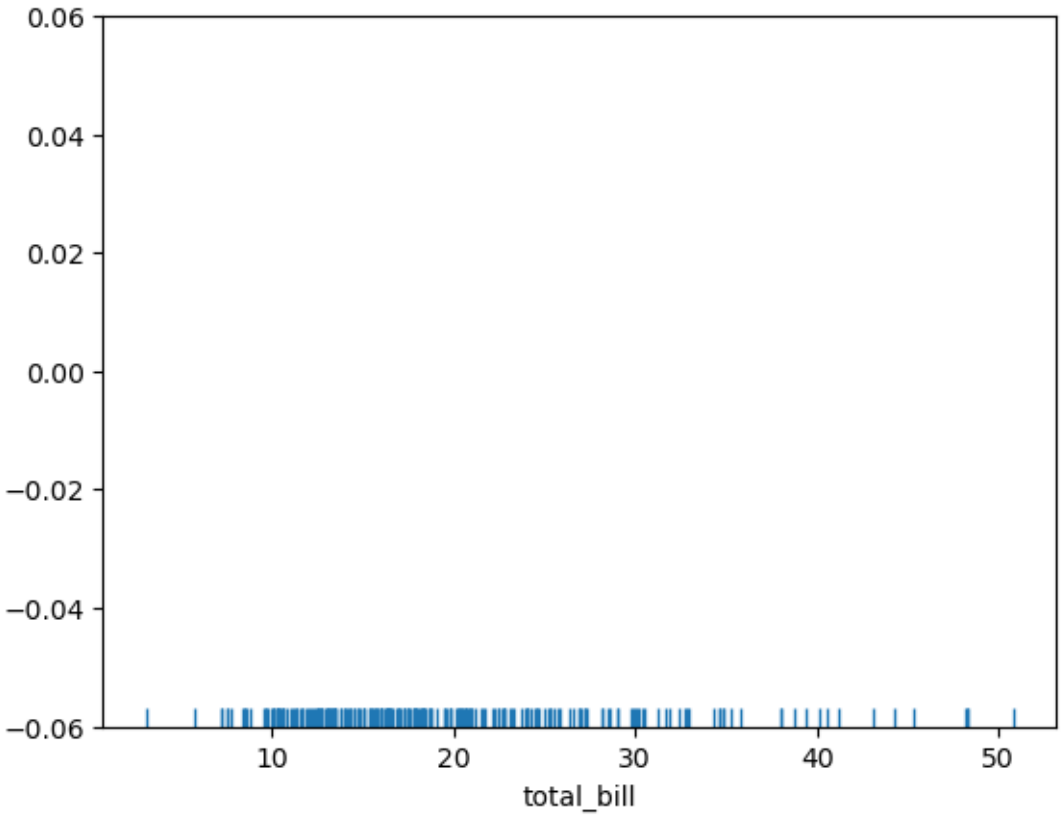
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[97]:
```
<seaborn.axisgrid.PairGrid at 0x1dc66f99810>
```



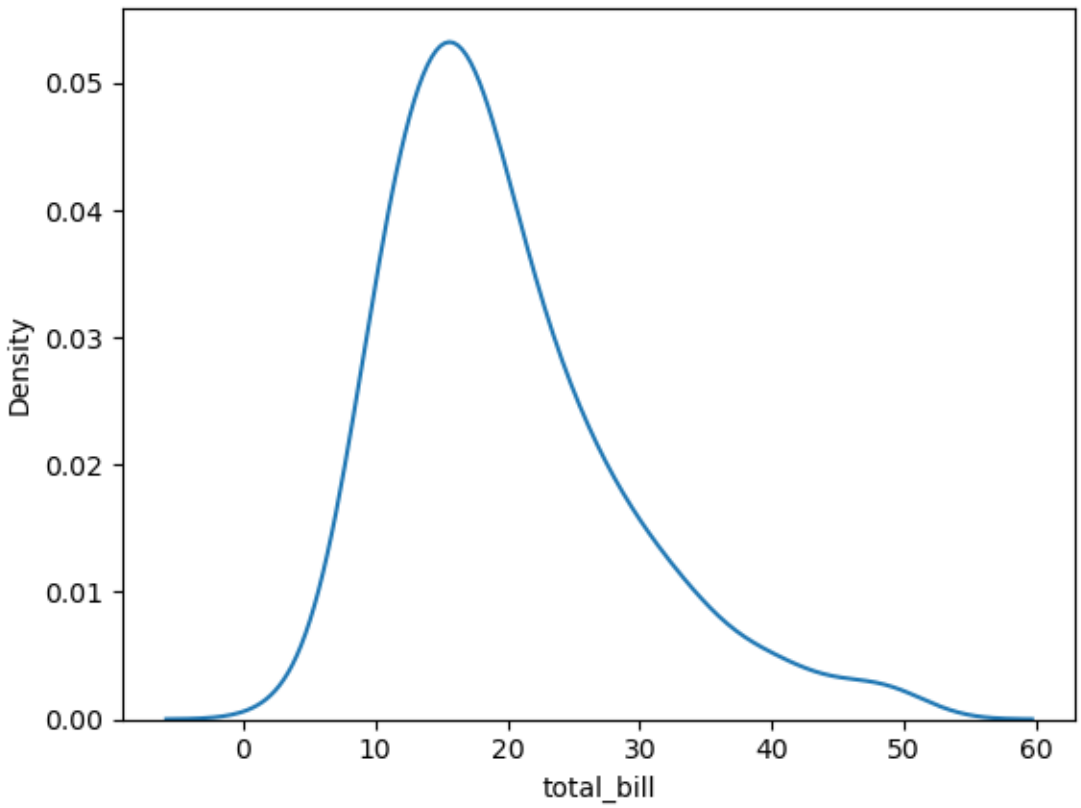
[72]:
```
sns.rugplot(tips['total_bill'])
```

[72]: <Axes: xlabel='total_bill'>



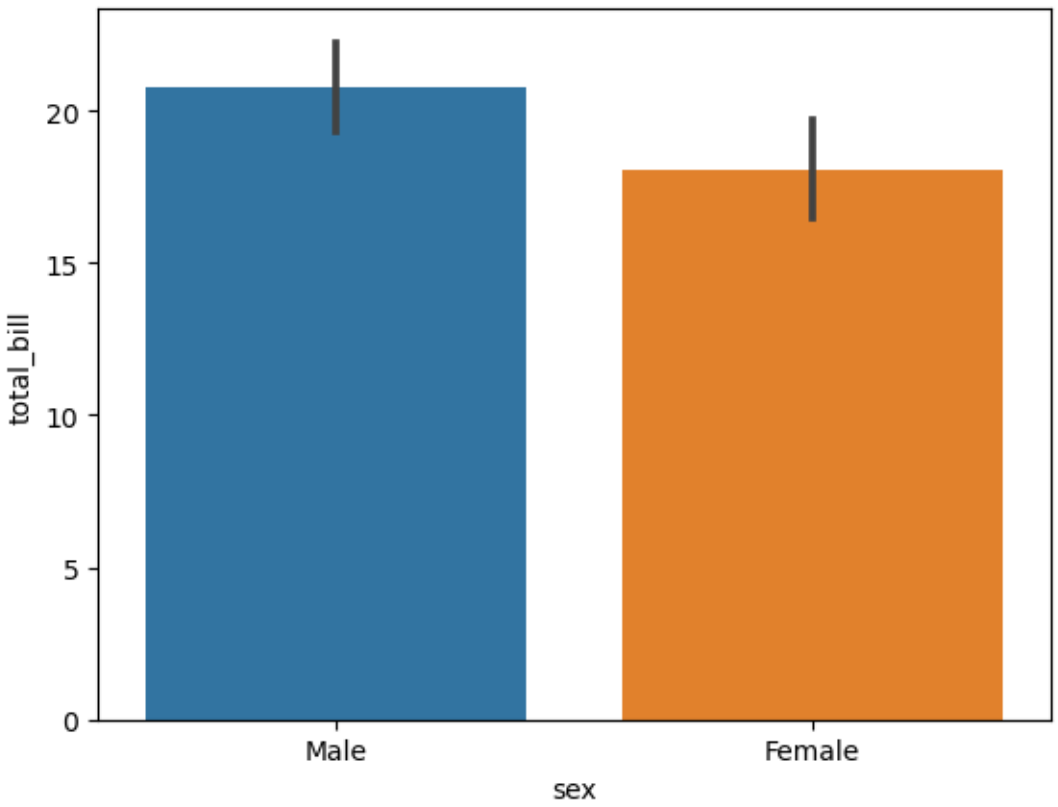
[73]:
```
sns.kdeplot(tips['total_bill'])
```

[73]:
```
<Axes: xlabel='total_bill', ylabel='Density'>
```



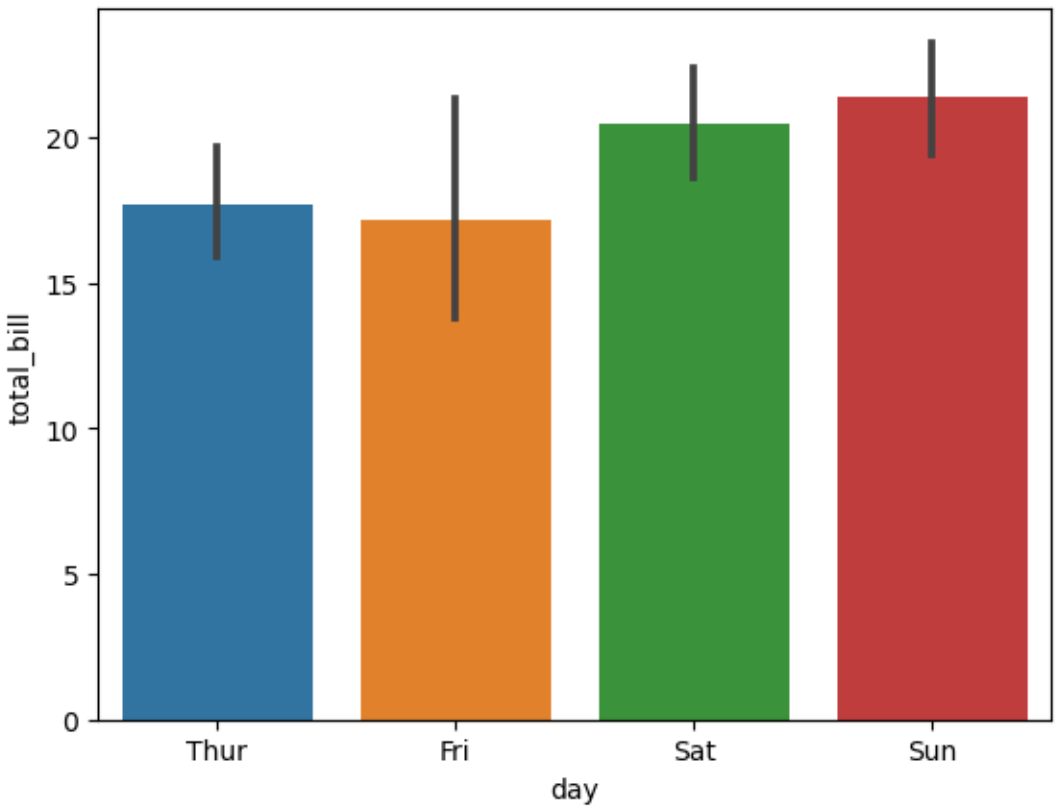
[98]:
```
sns.barplot(x='sex',y='total_bill',data=tips)
```

[98]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```



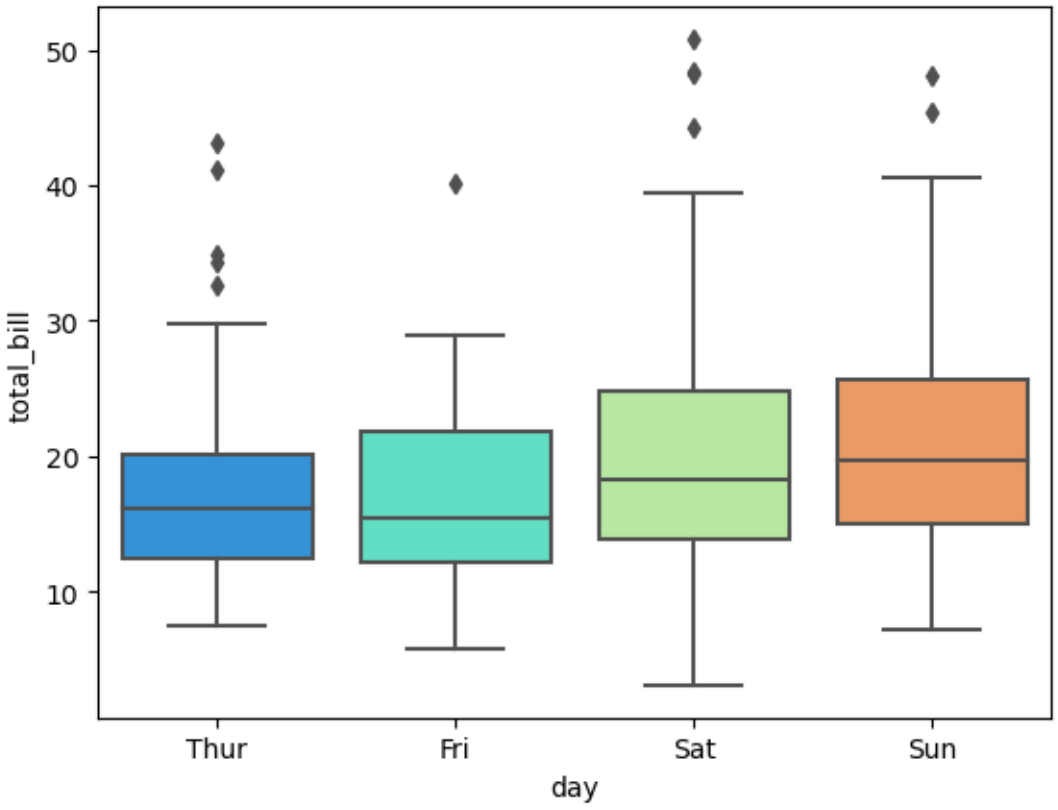
[99]:
```
sns.barplot(x='day',y='total_bill',data=tips)
```

[99]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



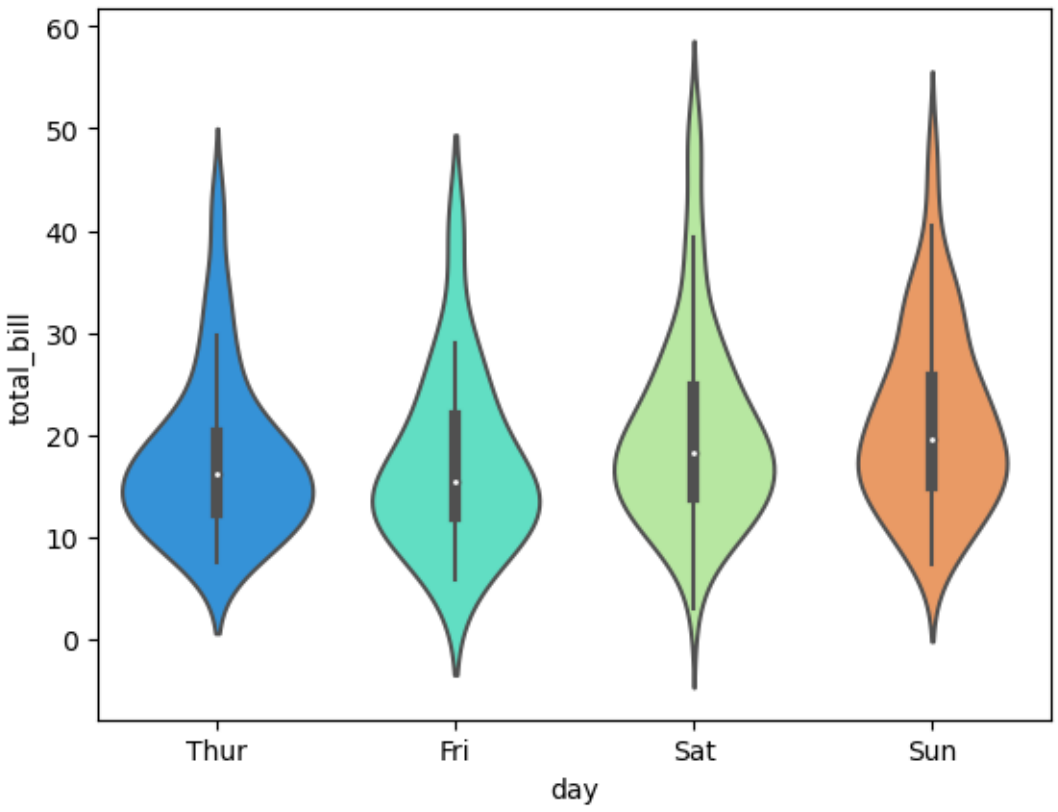
[102]:
```
sns.boxplot(x='day',y='total_bill',data=tips,palette='rainbow')
```

[102]: <Axes: xlabel='day', ylabel='total_bill'>



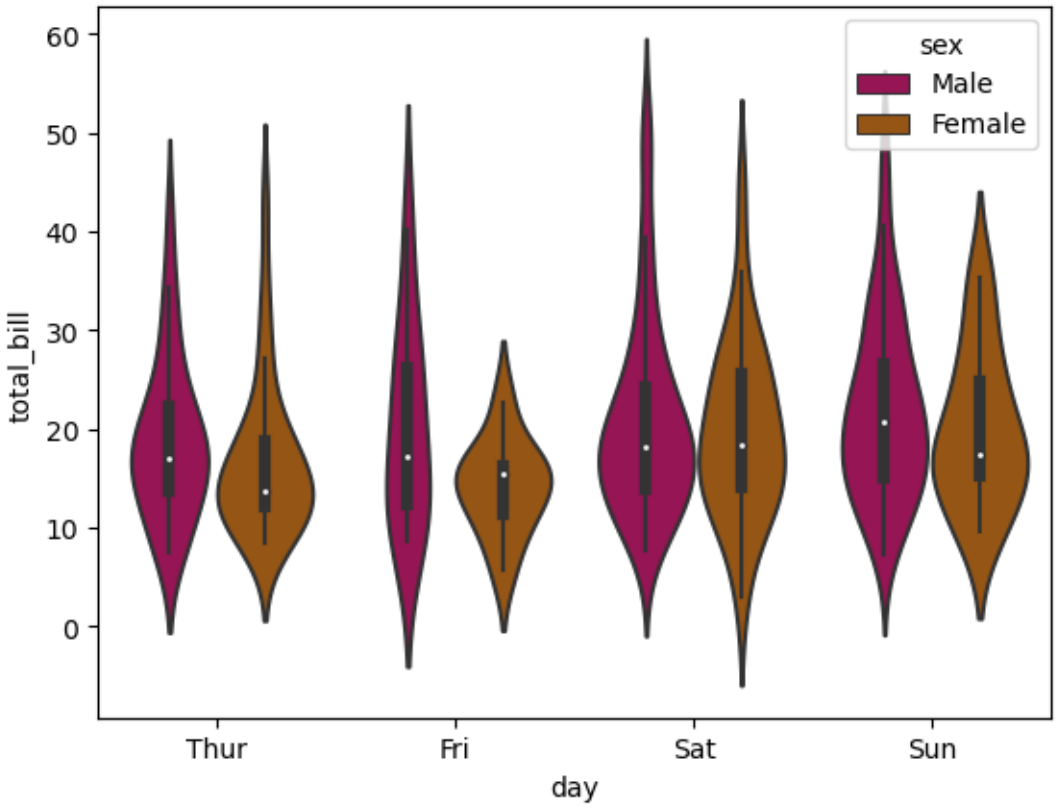
[109]:
```
sns.violinplot(x='day',y='total_bill',data=tips,palette='rainbow')
#violinplot contains density while boxplot doesnt(kde is also included)
```

[109]: <Axes: xlabel='day', ylabel='total_bill'>



[113]:
```
sns.violinplot(x='day',y='total_bill',data=tips,palette='brg',hue='sex')
```

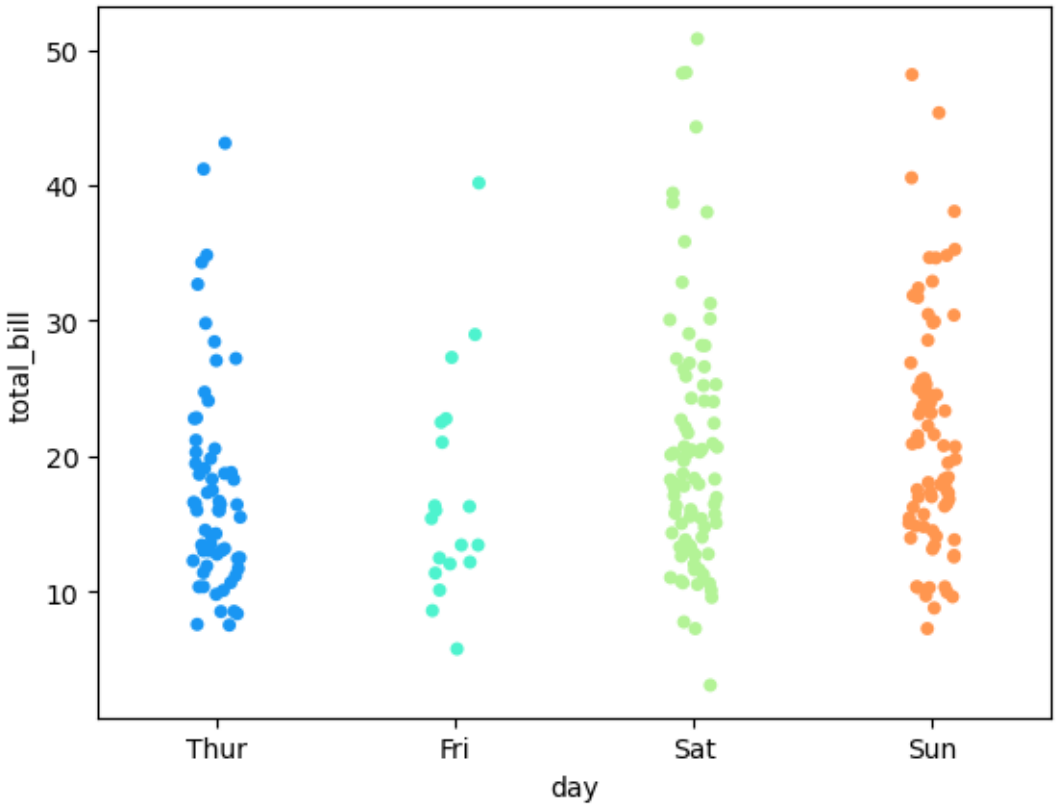
[113]: <Axes: xlabel='day', ylabel='total_bill'>



[110]:
```
sns.stripplot(x='day',y='total_bill',data=tips,palette='rainbow')
```

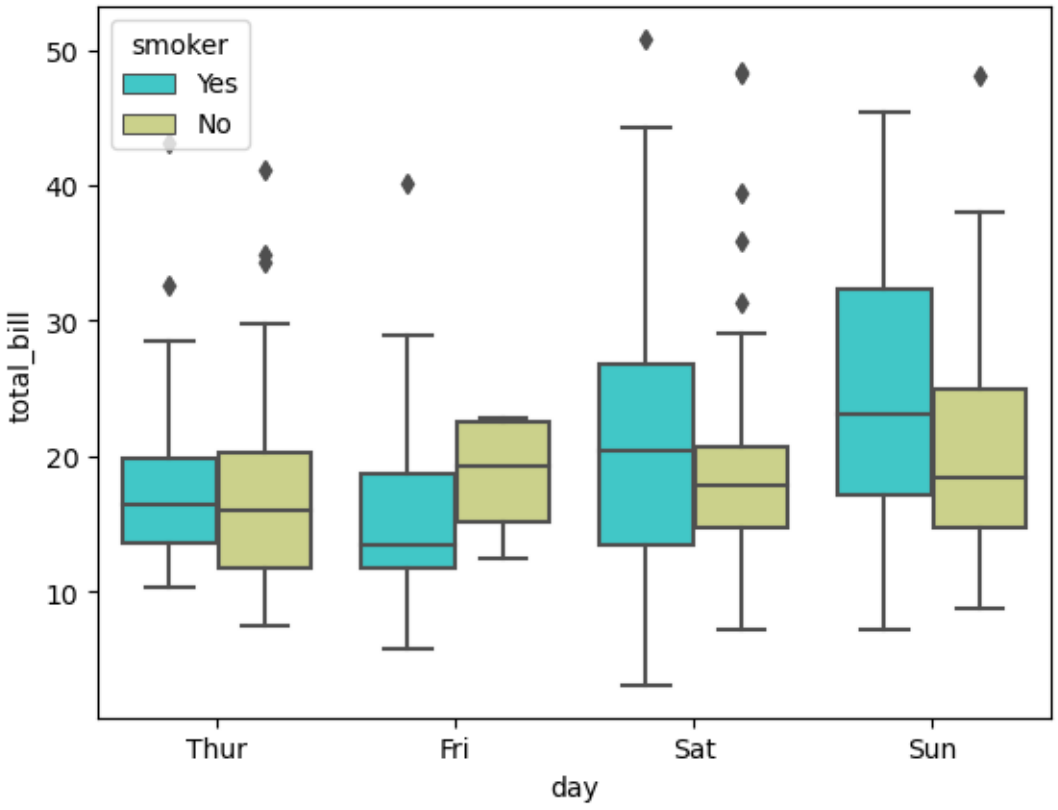
```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\652657083.py:1: FutureWarning:
Passing `palette` without assigning `hue` is deprecated.
  sns.stripplot(x='day',y='total_bill',data=tips,palette='rainbow')
```

[110]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



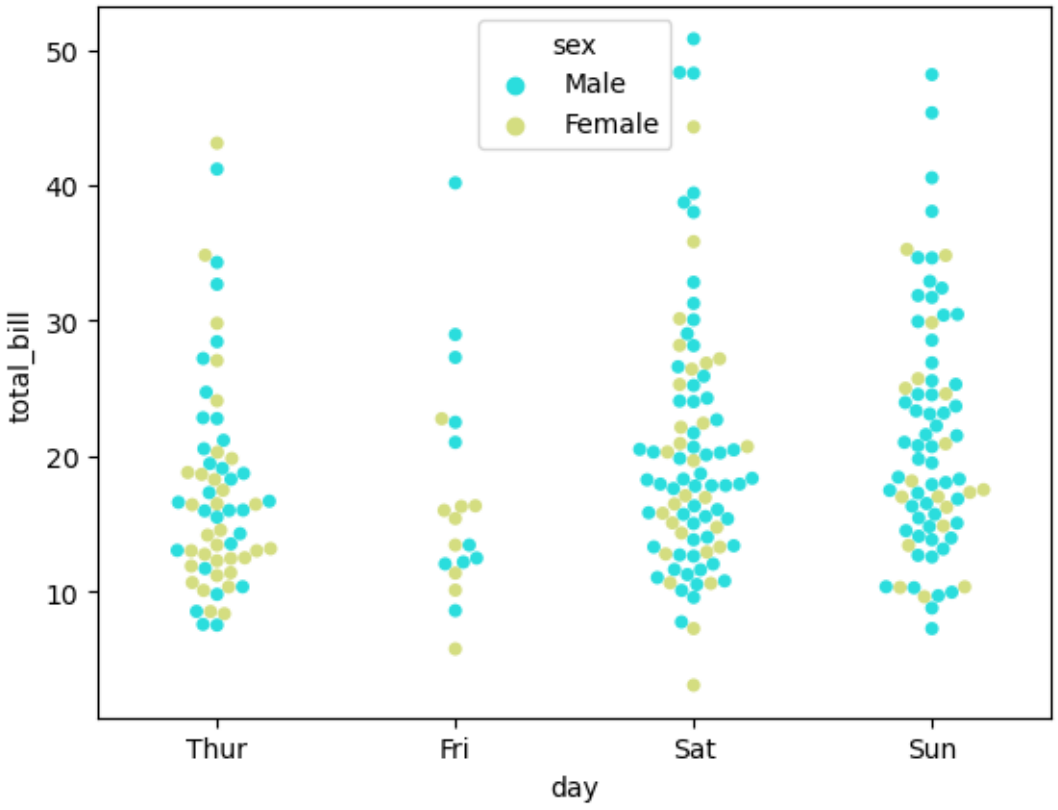
[103]:
```
sns.boxplot(x='day',y='total_bill',data=tips,hue='smoker',palette='rainbow')
```

[103]: <Axes: xlabel='day', ylabel='total_bill'>



[114]:
```
sns.swarmplot(x='day',y='total_bill',data=tips,palette='rainbow',hue='sex')
```

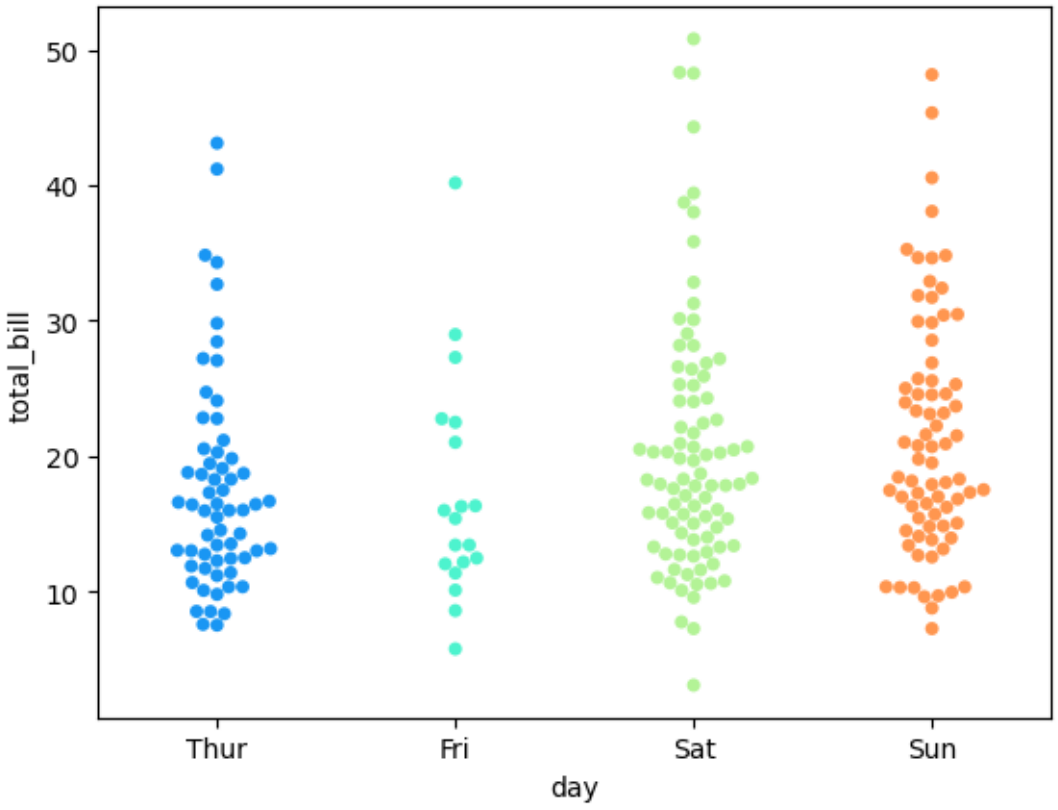
[114]: <Axes: xlabel='day', ylabel='total_bill'>



[115]:
```
sns.swarmplot(x='day',y='total_bill',data=tips,palette='rainbow')
```

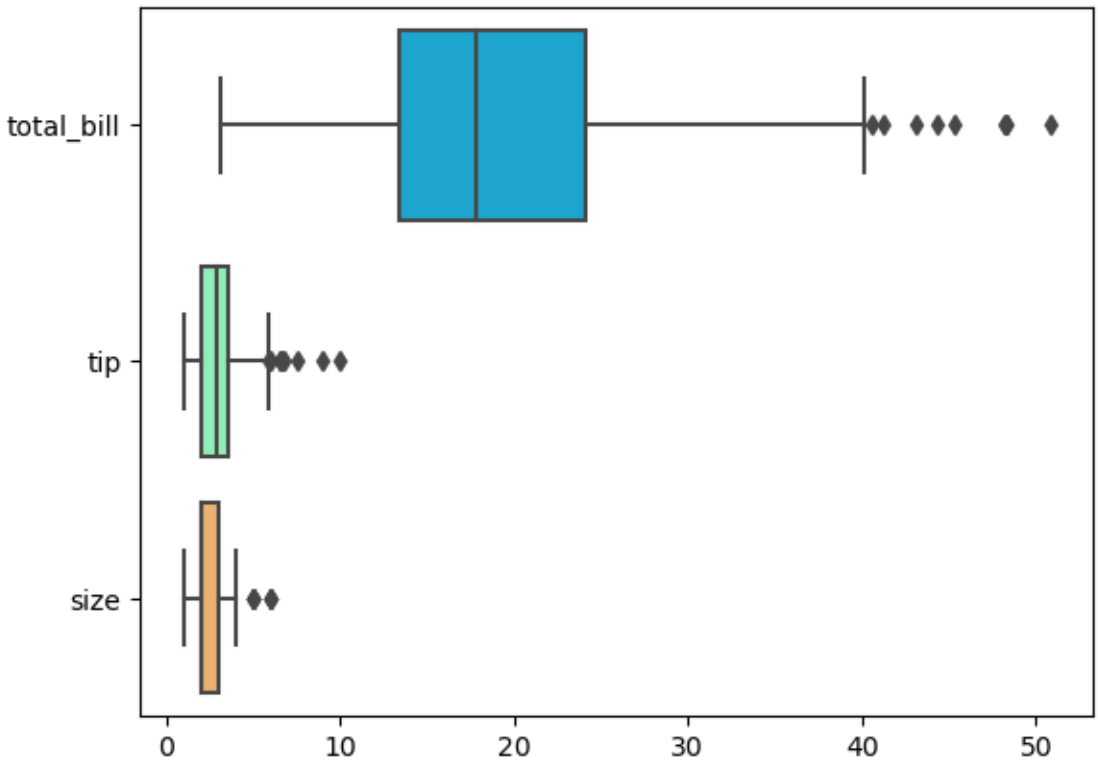
```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\657213969.py:1: FutureWarning:
Passing `palette` without assigning `hue` is deprecated.
  sns.swarmplot(x='day',y='total_bill',data=tips,palette='rainbow')
```

[115]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



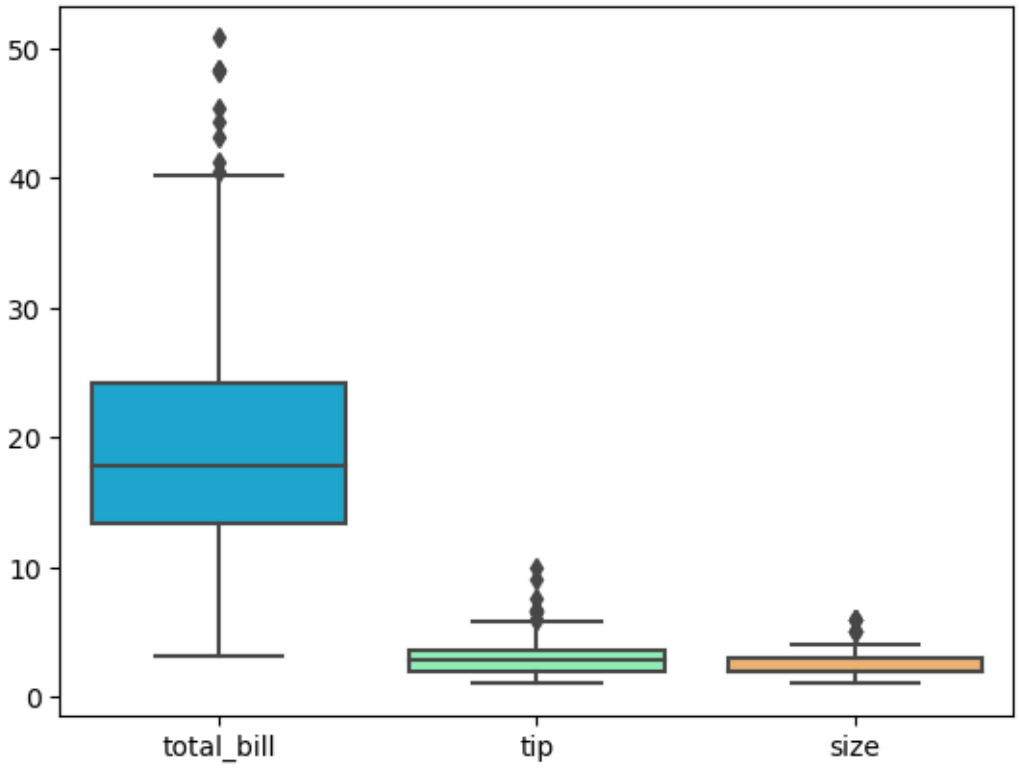
[105]:
```
sns.boxplot(data=tips,palette='rainbow',orient='h')
```

[105]: <Axes: >



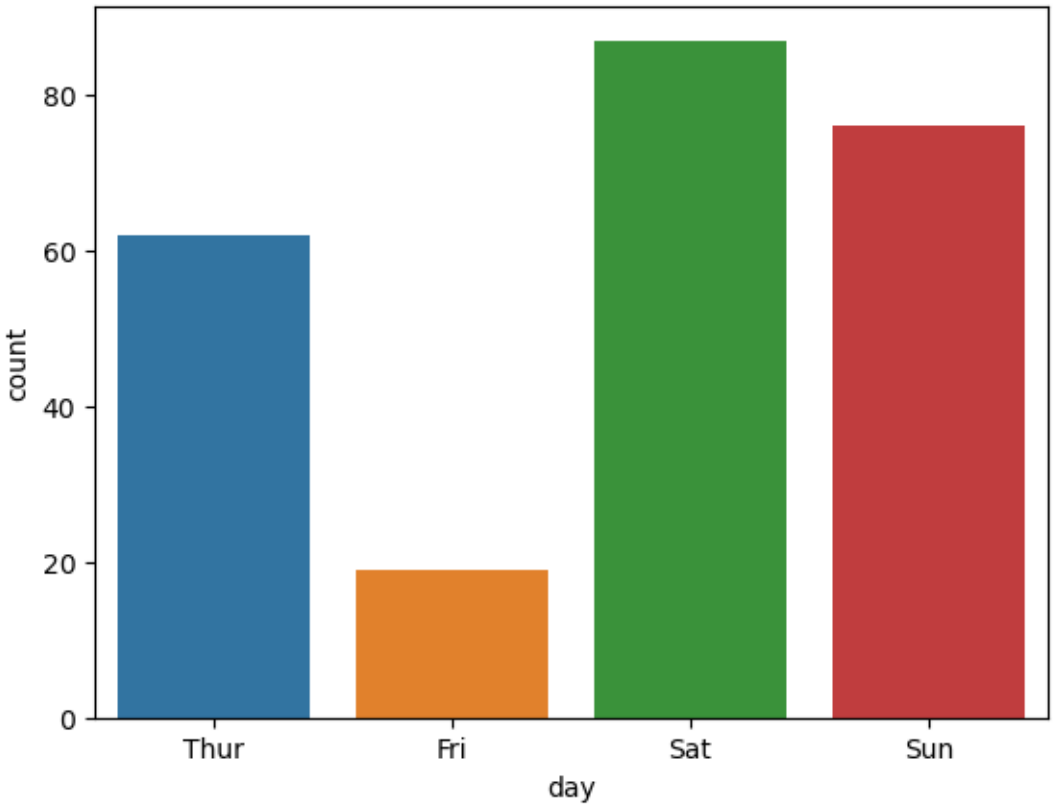
[106]:
```
sns.boxplot(data=tips,palette='rainbow',orient='v')
```

[106]: <Axes: >



[100]:
```
sns.countplot(x='day',data=tips)
```

[100]:
```
<Axes: xlabel='day', ylabel='count'>
```



[74]:
```
tips.head()
```

[74]:

	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

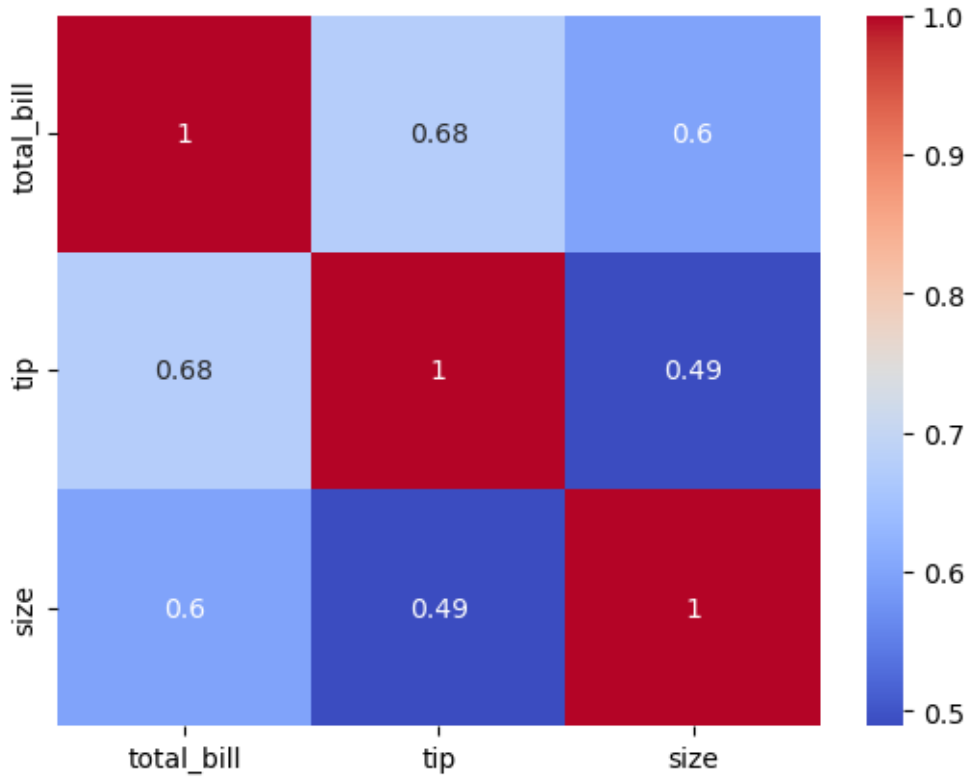
[93]:
```
tips.corr(numeric_only=True)
```

[93]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

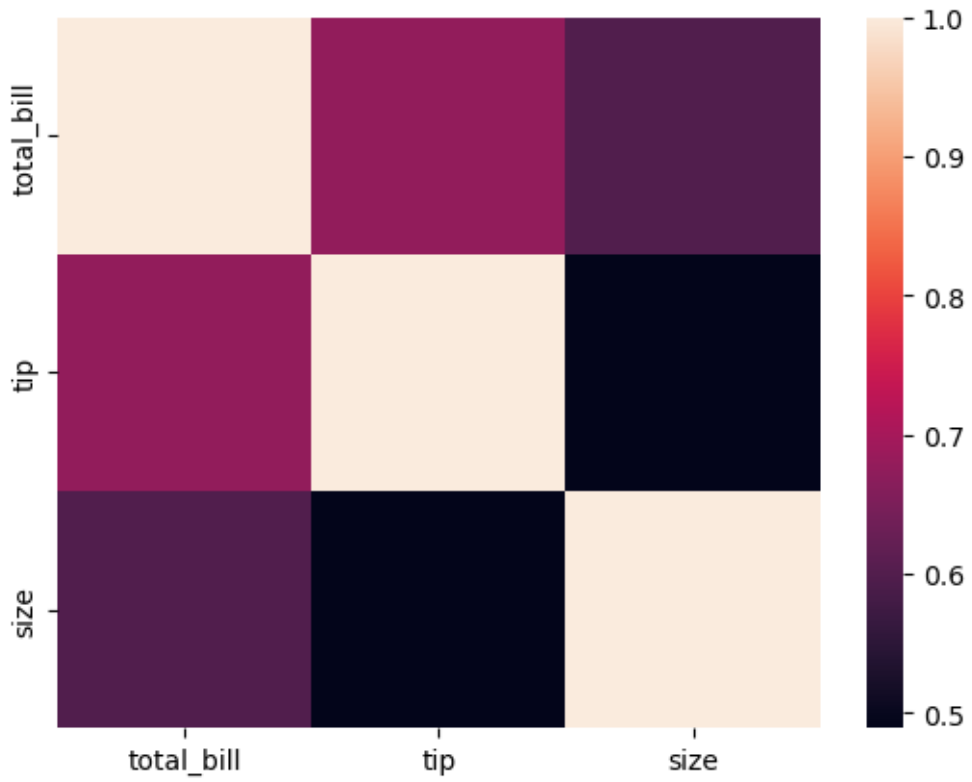
[94]:
```
sns.heatmap(tips.corr(numeric_only=True),cmap='coolwarm',annot=True)
```

[94]: <Axes: >



[95]:
```
sns.heatmap(tips.corr(numeric_only=True))
```

[95]: <Axes: >



[75]:
```
flight=sns.load_dataset('flights')
flight.head()
```

[75]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121

[76]:
```
pvflight = flight.pivot_table(index='month', columns='year',␣
  ↪values='passengers')
pvflight
```

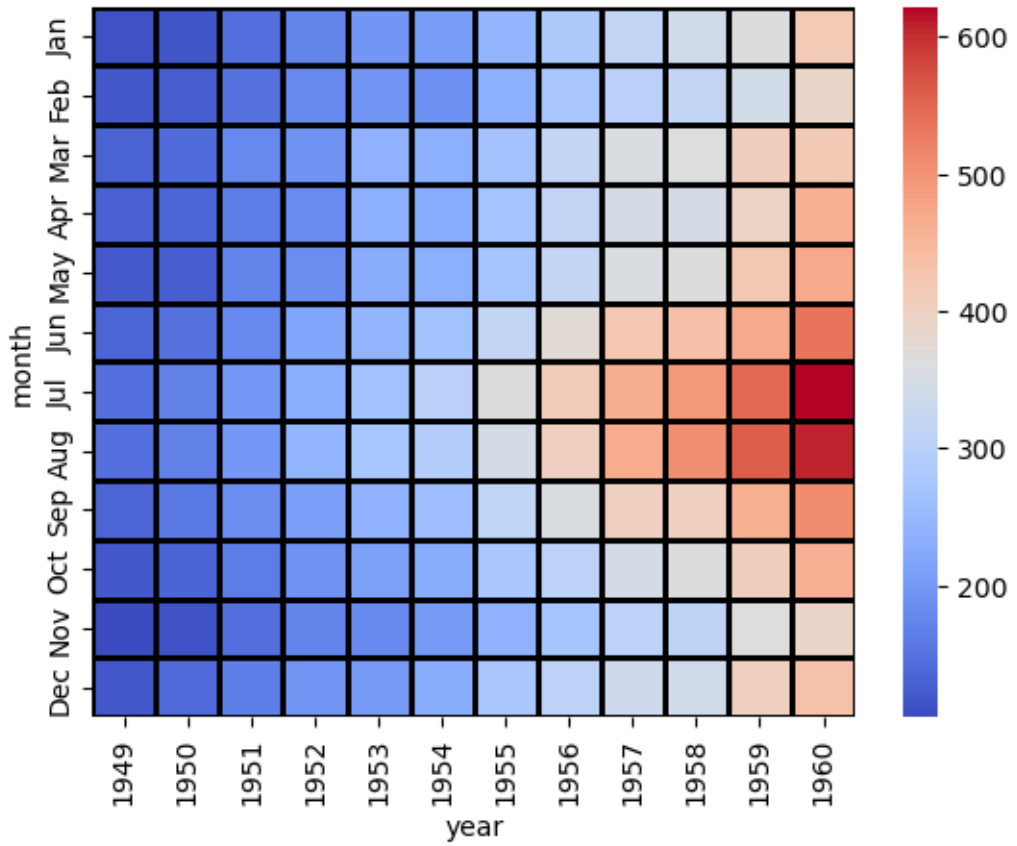
[76]:

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535

Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

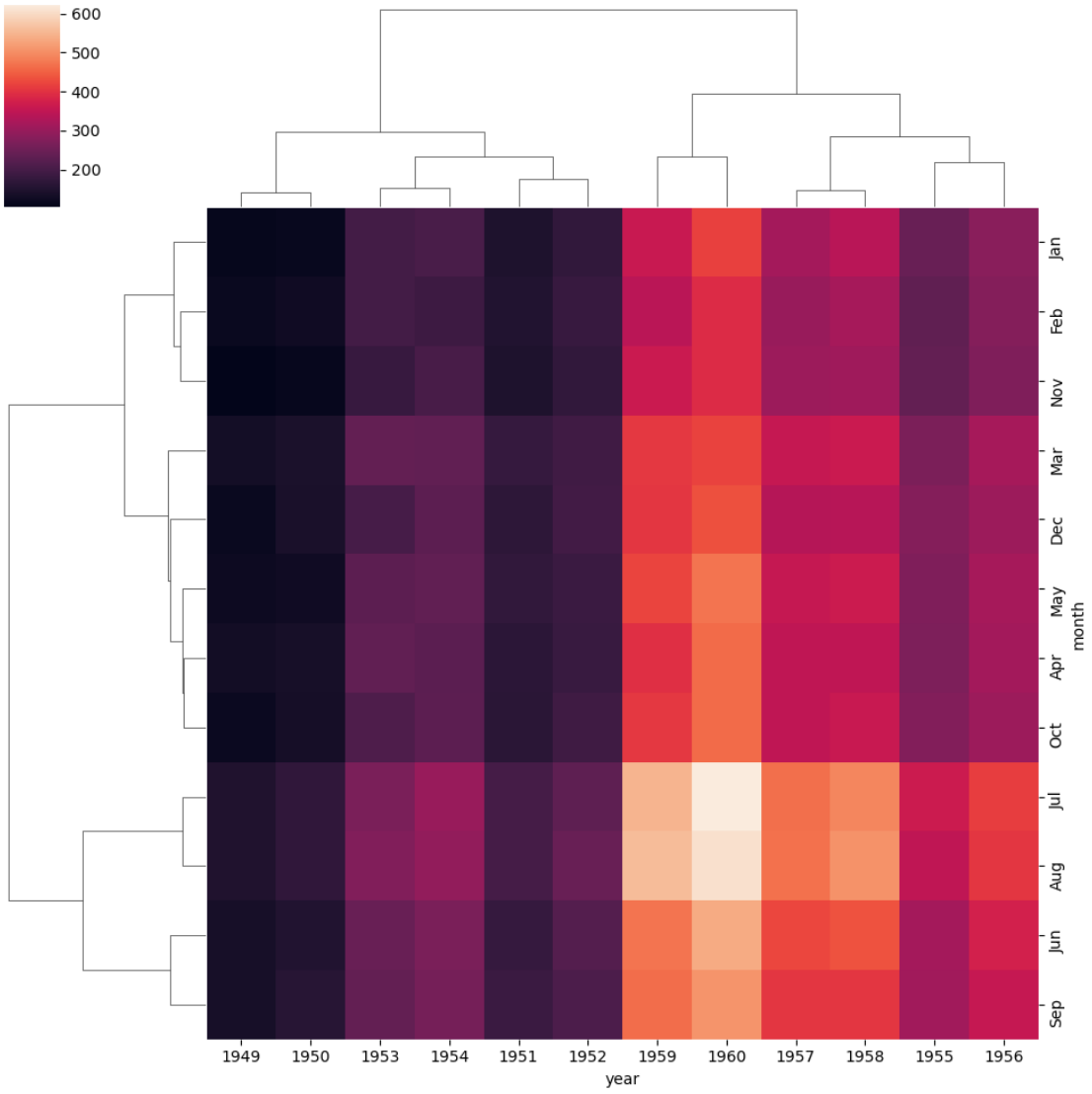
[77]:
```
sns.heatmap(pvflight,cmap='coolwarm',linecolor='black',linewidth=1)
```

[77]:
```
<Axes: xlabel='year', ylabel='month'>
```



[78]:
```
sns.clustermap(pvflight)
```

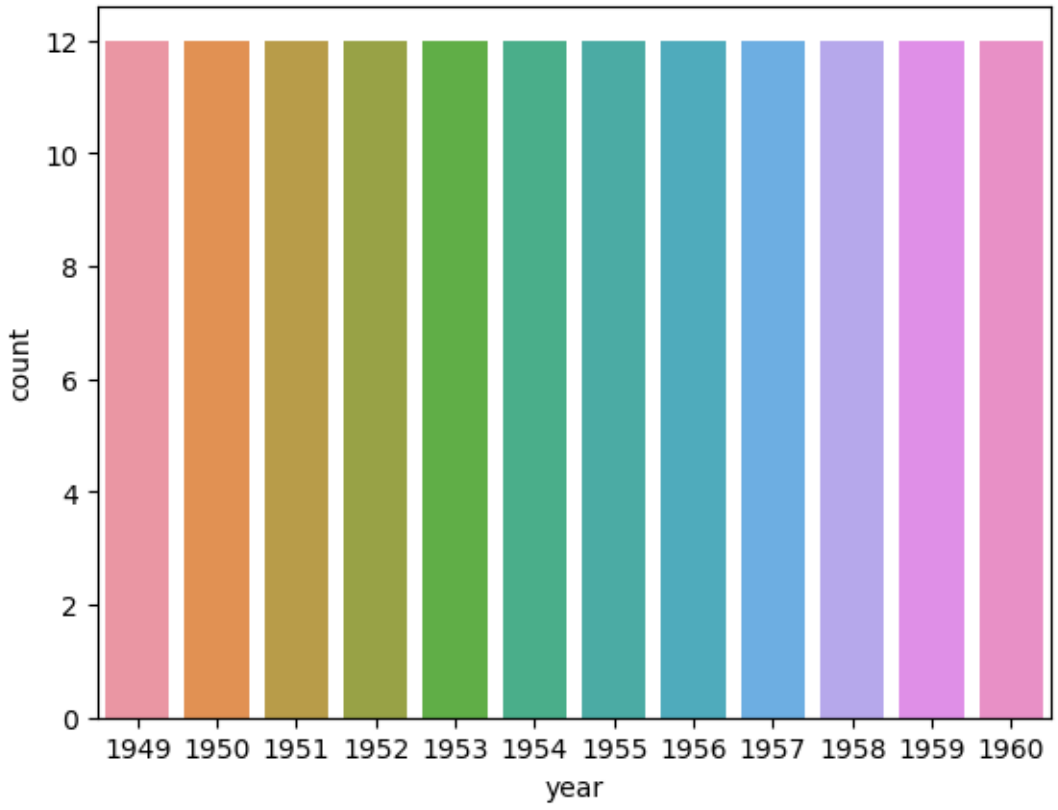
[78]:
```
<seaborn.matrix.ClusterGrid at 0x1dc63859010>
```



pairplot will plot pairwaise relationship across an entire dataframe.(for the numerical columns) ans support a color, hue argument(for categorical columns)

[96]:
```
sns.countplot(pvflight)
```

[96]:
```
<Axes: xlabel='year', ylabel='count'>
```



[116]:
```
titanic=sns.load_dataset('titanic')
titanic.head()
```

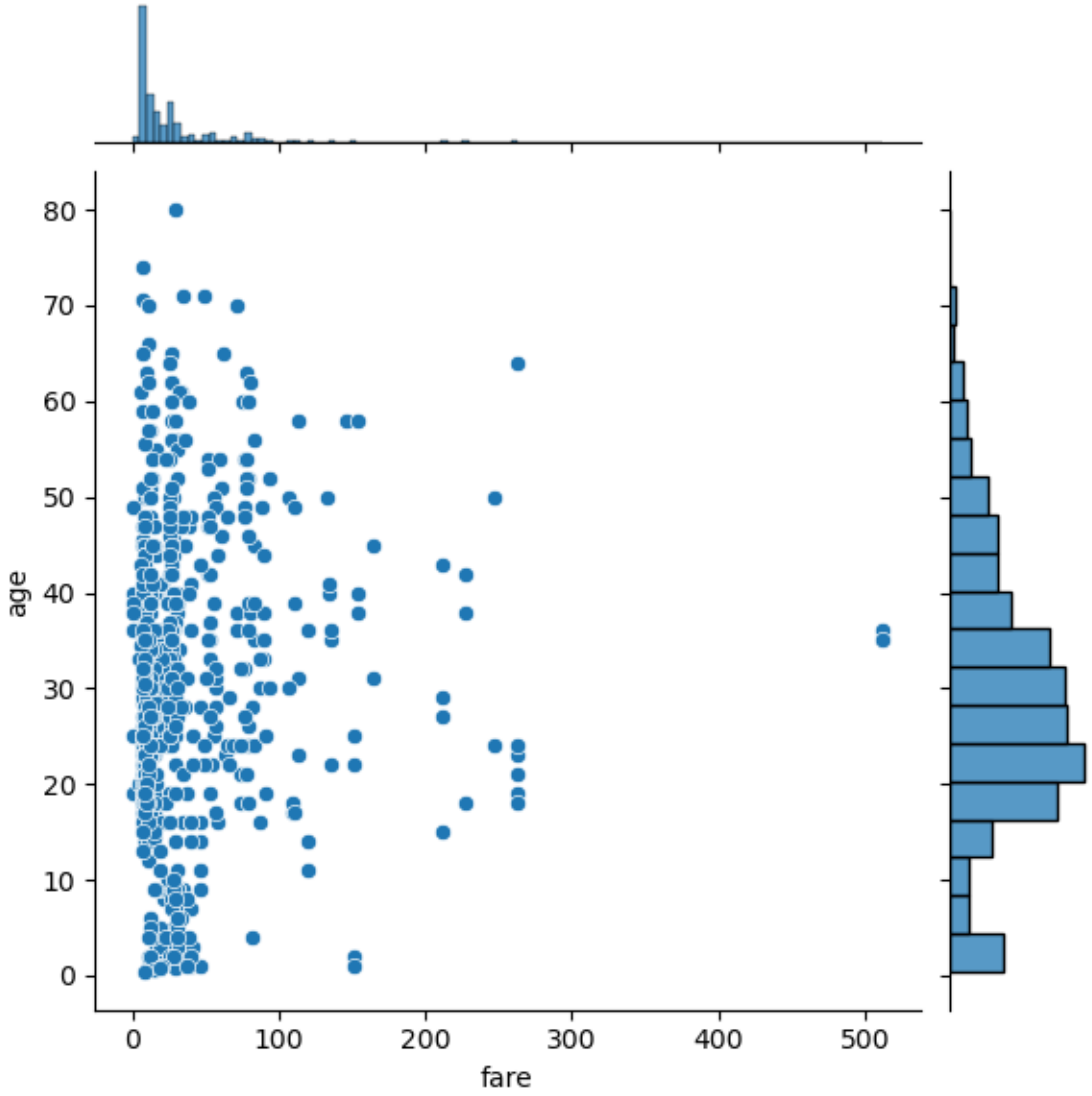
[116]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	\
0	0	3	male	22.0	1	0	7.2500	S	Third	
1	1	1	female	38.0	1	0	71.2833	C	First	
2	1	3	female	26.0	0	0	7.9250	S	Third	
3	1	1	female	35.0	1	0	53.1000	S	First	
4	0	3	male	35.0	0	0	8.0500	S	Third	

	who	adult_male	deck	embark_town	alive	alone
0	man	True	NaN	Southampton	no	False
1	woman	False	C	Cherbourg	yes	False
2	woman	False	NaN	Southampton	yes	True
3	woman	False	C	Southampton	yes	False
4	man	True	NaN	Southampton	no	True

[117]:
```
sns.jointplot(x='fare', y='age', data=titanic)
```

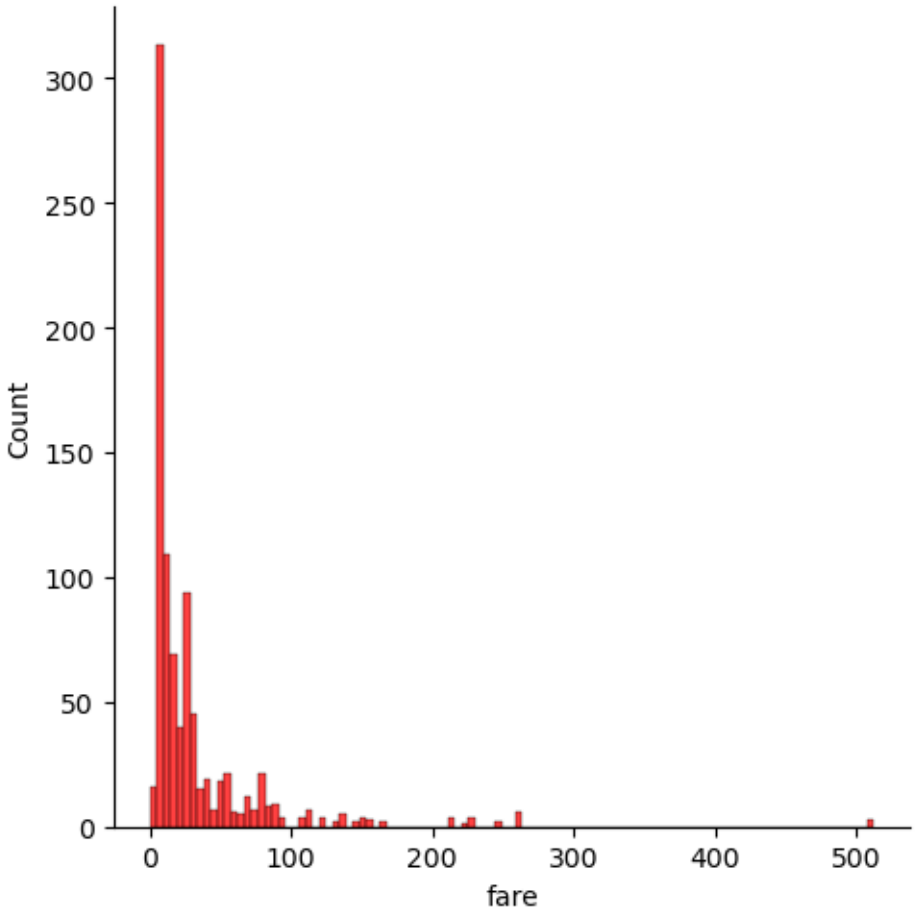
[117]:
```
<seaborn.axisgrid.JointGrid at 0x1dc671f2cd0>
```



[125]:
```
sns.displot(titanic['fare'],color='red')
```

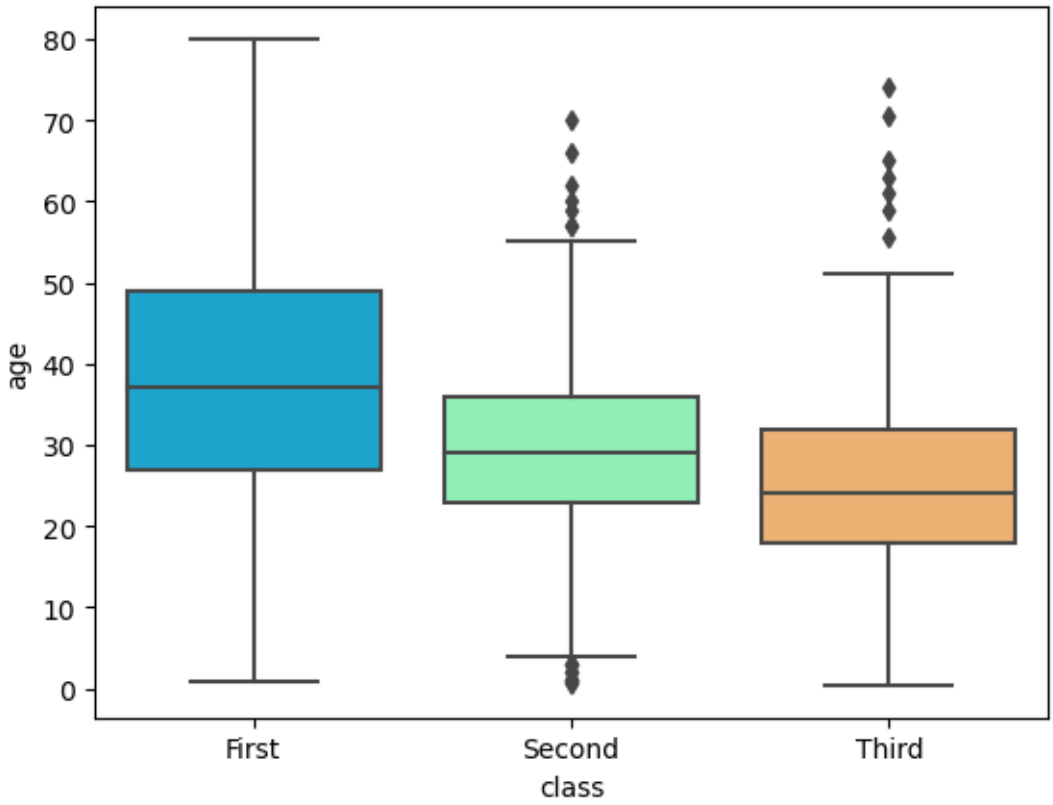
```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

[125]: <seaborn.axisgrid.FacetGrid at 0x1dc6b6c6550>



[126]:
```
sns.boxplot(x='class',y='age',data=titanic,palette='rainbow')
```

[126]:
```
<Axes: xlabel='class', ylabel='age'>
```

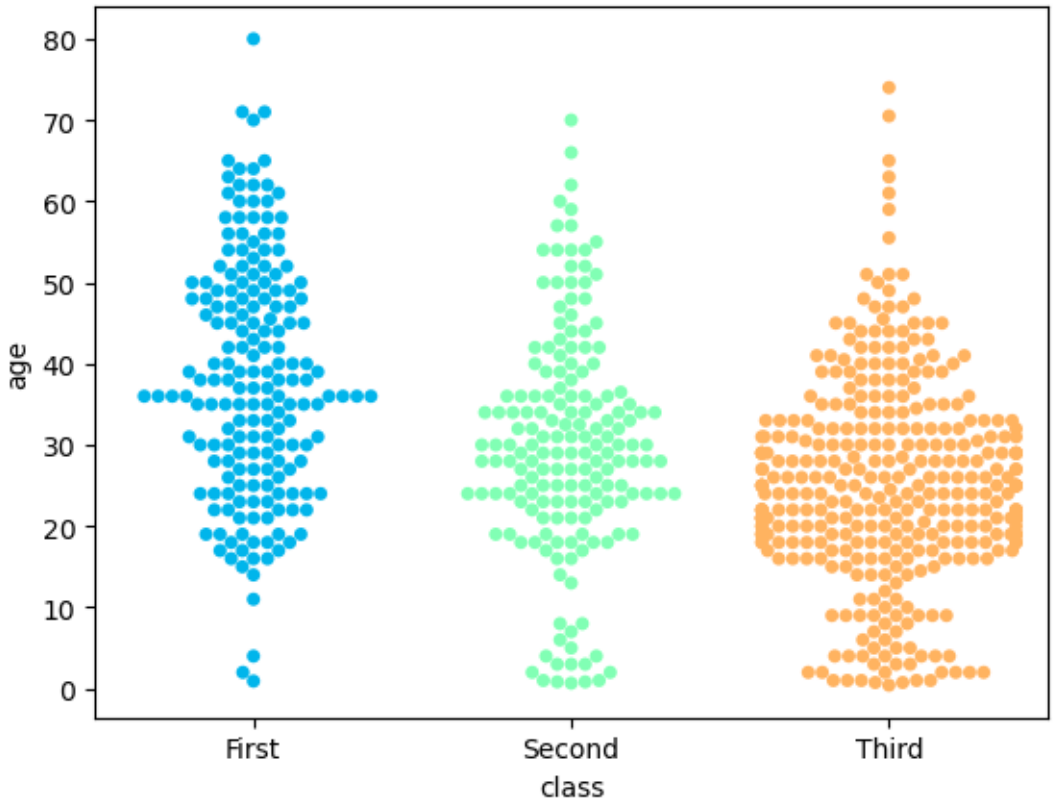


[129]:
```
sns.swarmplot(x='class',y='age',data=titanic,palette='rainbow')
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_19508\173160962.py:1: FutureWarning:
Passing `palette` without assigning `hue` is deprecated.
  sns.swarmplot(x='class',y='age',data=titanic,palette='rainbow')
```

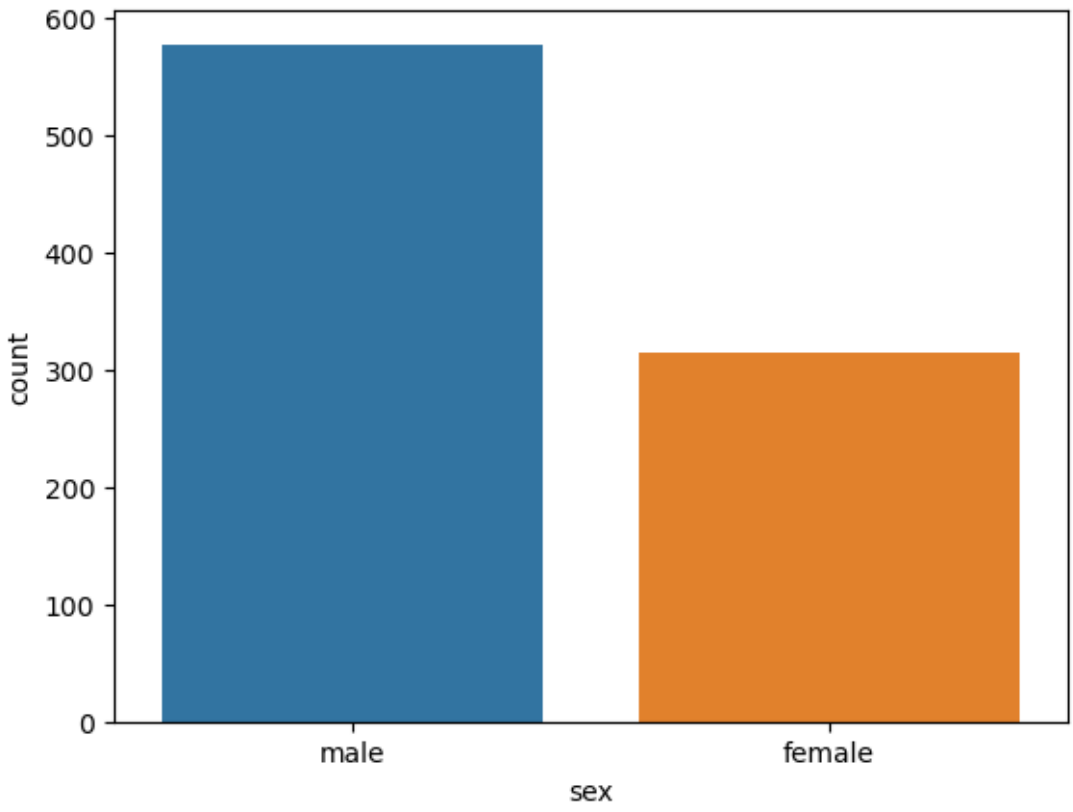
[129]:
```
<Axes: xlabel='class', ylabel='age'>
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\categorical.py:3544:
UserWarning: 15.2% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



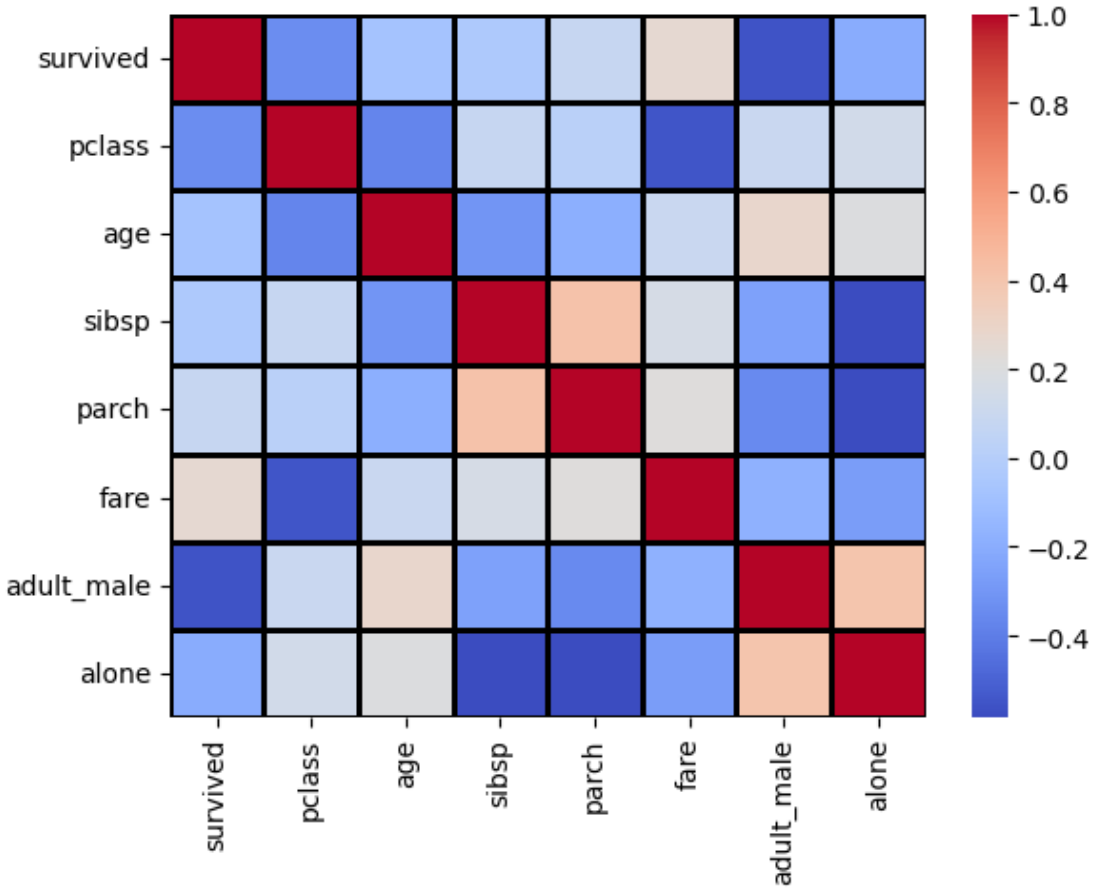
[134]:
```
sns.countplot(x=titanic['sex'],data=titanic)
```

[134]:
```
<Axes: xlabel='sex', ylabel='count'>
```



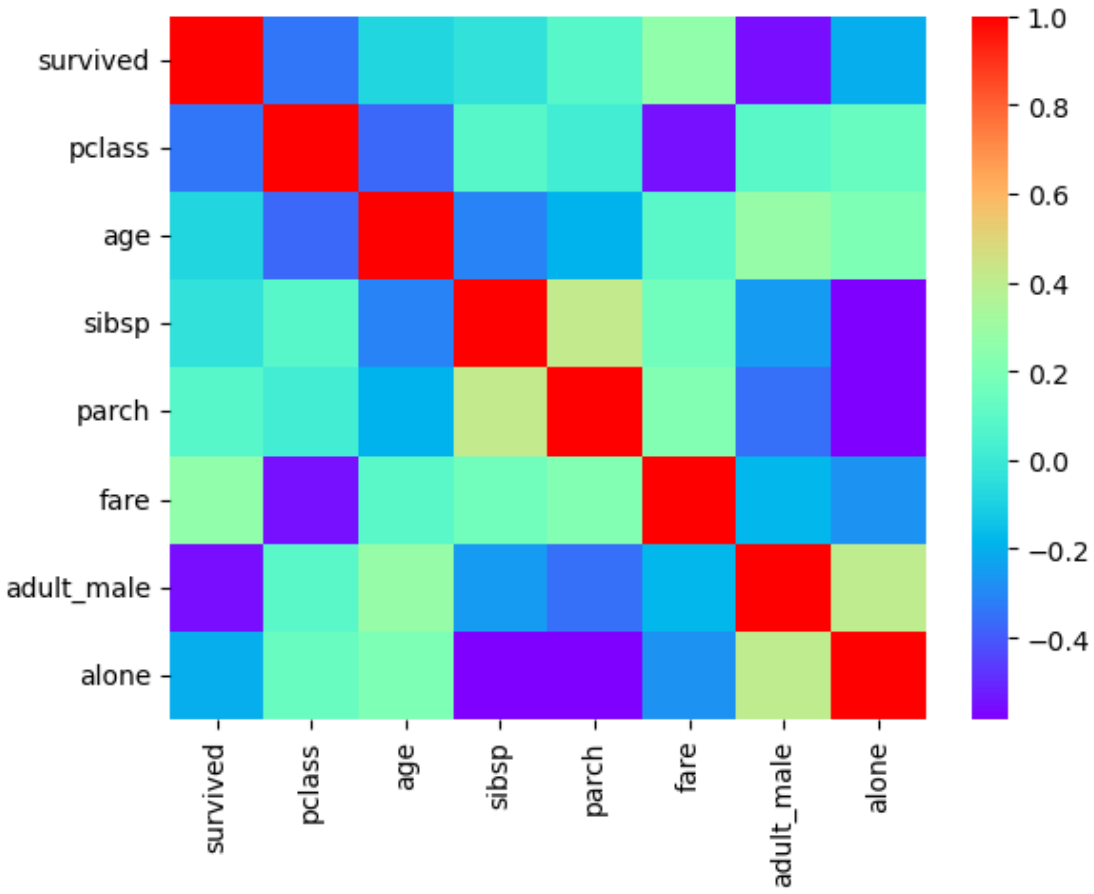
[138]:
```
sns.heatmap(titanic.
↪corr(numeric_only=True),cmap='coolwarm',linecolor='black',linewidth=1)
```

[138]: <Axes: >



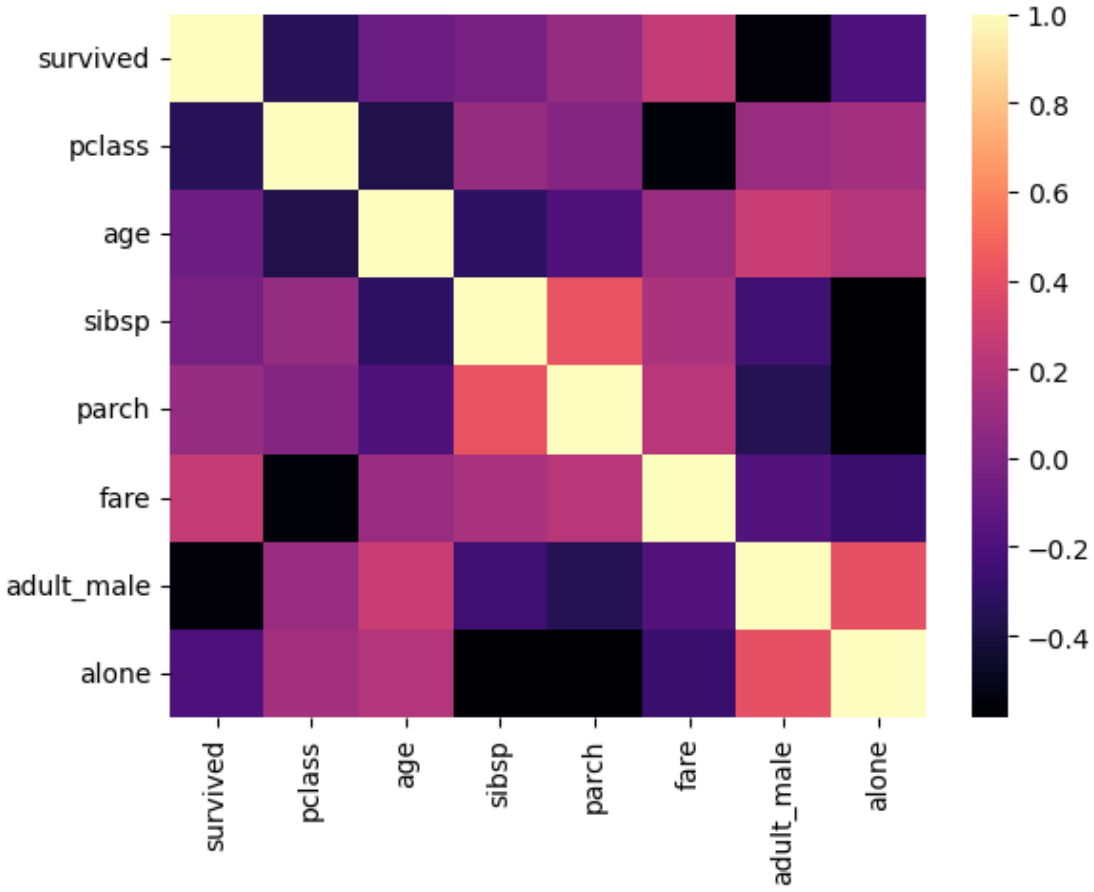
[139]:
```
sns.heatmap(titanic.corr(numeric_only=True),cmap='rainbow')
```

[139]: <Axes: >



[140]:
```
sns.heatmap(titanic.corr(numeric_only=True),cmap='magma')
```

[140]: <Axes: >


survived
pclass
age
sibsp
parch
fare
adult_male
alone
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

[ ]: