

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

```
In [2]: 1 import numpy as np
```

```
In [3]: 1 x=np.arange(0,10)
2 y=np.arange(11,21)
```

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

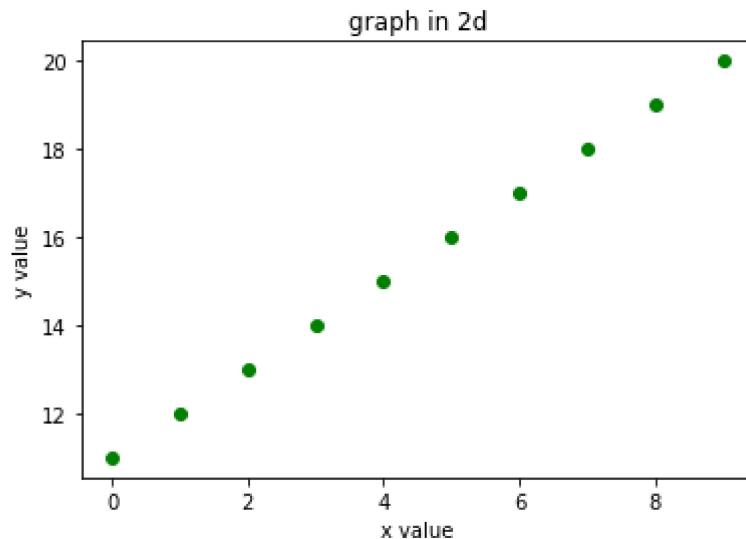
```
Out[4]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

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

```
Out[5]: array([11, 12, 13, 14, 15, 16, 17, 18, 19, 20])
```

scatter plot

```
In [6]: 1 plt.scatter(x,y,c='g')
2 plt.xlabel('x value')
3 plt.ylabel('y value')
4 plt.title('graph in 2d')
5 plt.savefig('test.png')
```

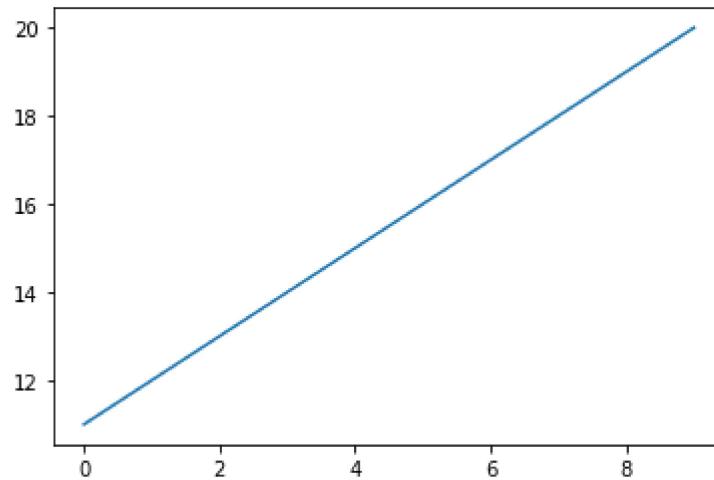


line plot

```
In [10]: 1 x=np.arange(0,10)
2 y=np.arange(11,21)
```

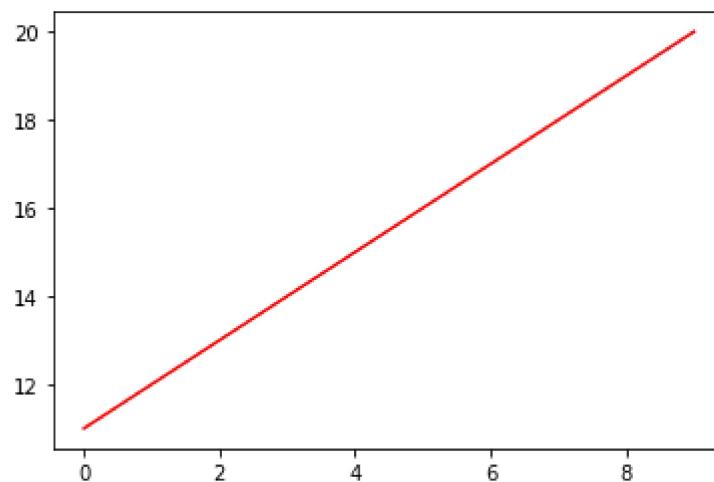
In [11]: 1 plt.plot(x,y)

Out[11]: [`<matplotlib.lines.Line2D at 0x6cc0550>`]



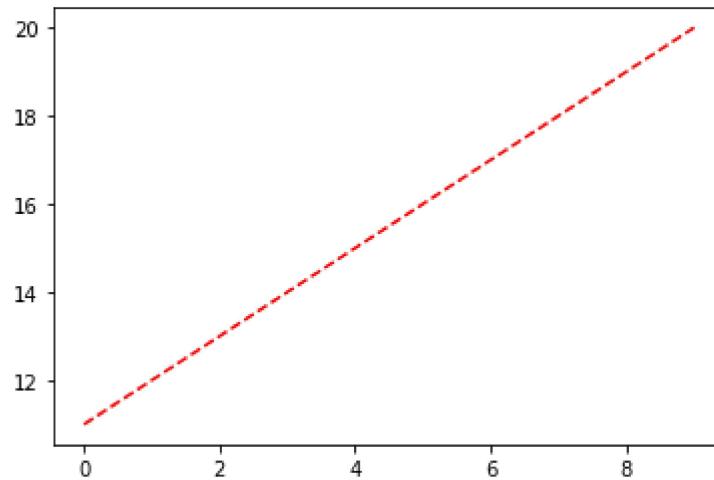
In [12]: 1 plt.plot(x,y,'r')

Out[12]: [`<matplotlib.lines.Line2D at 0x6d13340>`]



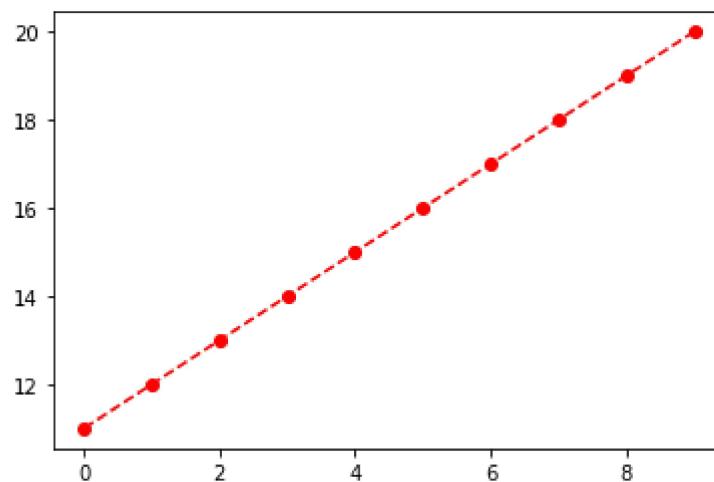
In [13]: 1 plt.plot(x,y, 'r--')

Out[13]: [`<matplotlib.lines.Line2D at 0x6d5ef70>`]



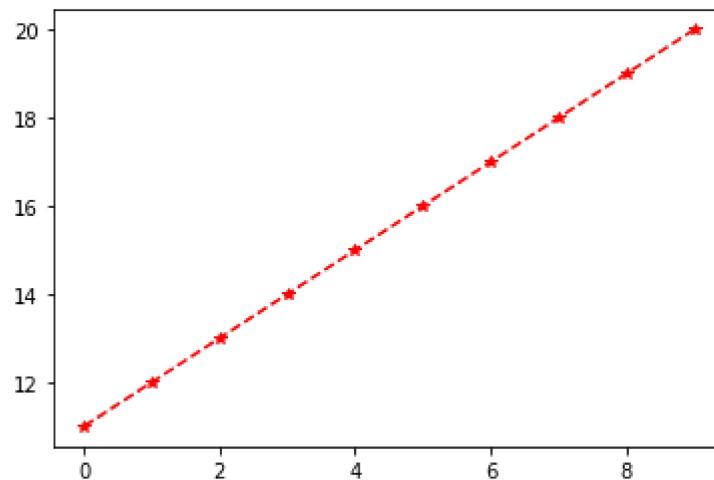
In [14]: 1 plt.plot(x,y, 'ro--')

Out[14]: [`<matplotlib.lines.Line2D at 0x6fa6c40>`]



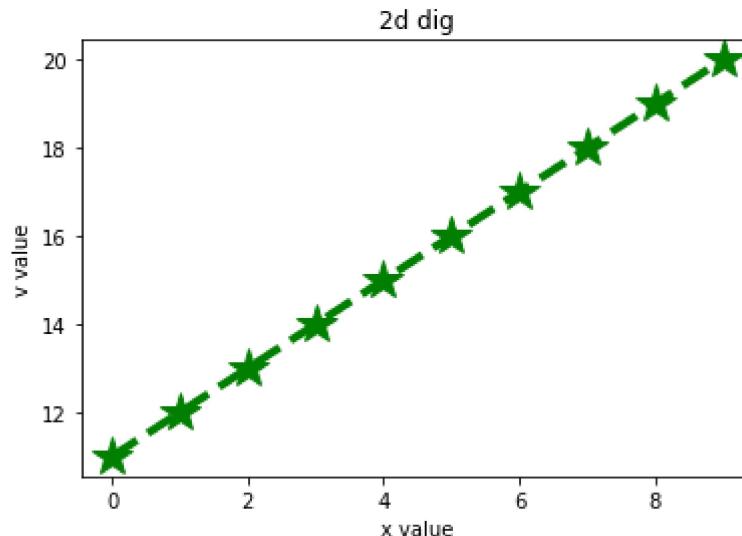
```
In [15]: 1 plt.plot(x,y,'r*--')
```

```
Out[15]: [<matplotlib.lines.Line2D at 0x6ffa940>]
```



```
In [19]: 1 plt.plot(x,y,'g*',linestyle='dashed',linewidth=4,markersize=20)
 2 plt.xlabel('x value')
 3 plt.ylabel('v value')
 4 plt.title('2d dig')
```

```
Out[19]: Text(0.5, 1.0, '2d dig')
```



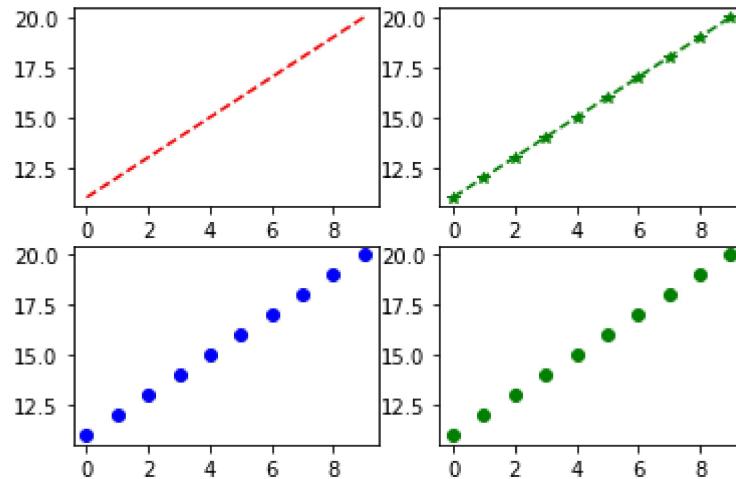
subplot

In [21]:

```

1 plt.subplot(2,2,1)
2 plt.plot(x,y,'r--')
3 plt.subplot(2,2,2)
4 plt.plot(x,y,'g*--')
5 plt.subplot(2,2,3)
6 plt.plot(x,y,'bo')
7 plt.subplot(2,2,4)
8 plt.plot(x,y,'go')
9 plt.savefig('subdig')

```



bar plot

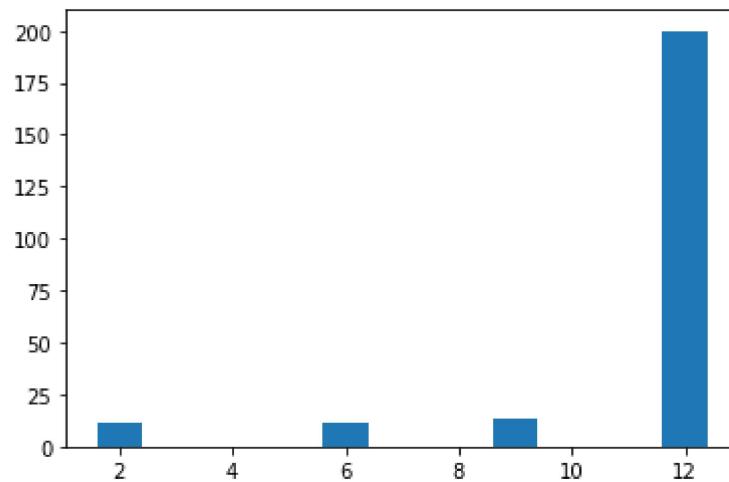
In [24]:

```

1 x=[2,6,9,12]
2 y=[11,11,13,200]
3 plt.bar(x,y)

```

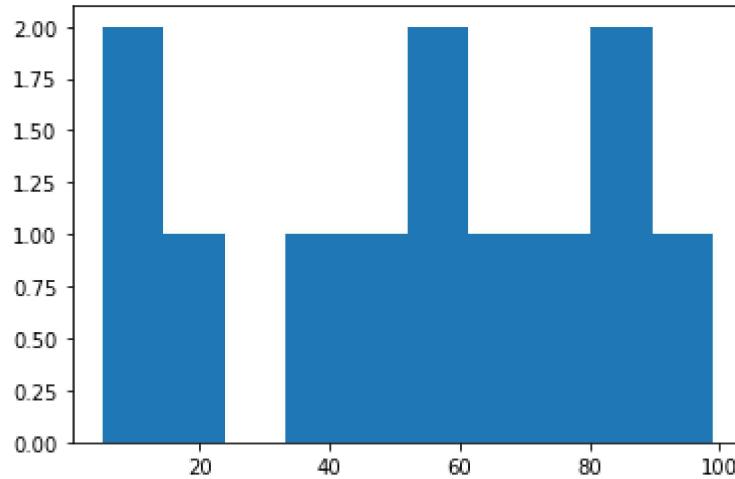
Out[24]: <BarContainer object of 4 artists>



histogram

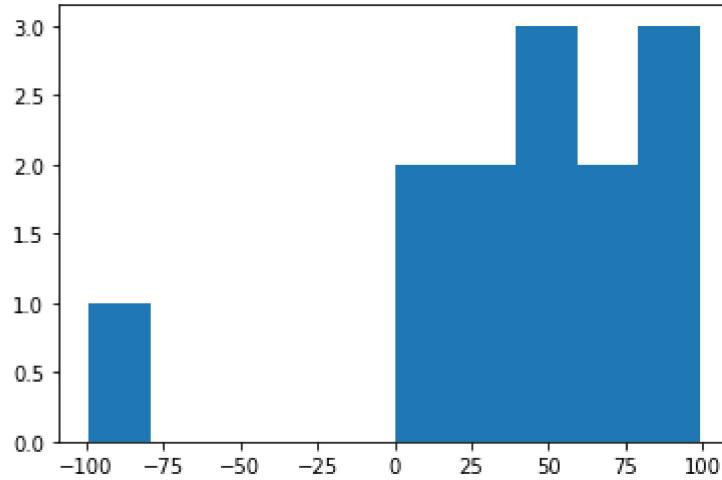
```
In [25]: 1 a=np.array([22,87,5,43,56,78,55,67,88,99,12,34])  
2 plt.hist(a)
```

```
Out[25]: (array([2., 1., 0., 1., 2., 1., 1., 2., 1.]),  
 array([ 5. , 14.4, 23.8, 33.2, 42.6, 52. , 61.4, 70.8, 80.2, 89.6, 99. ]),  
 <BarContainer object of 10 artists>)
```



```
In [28]: 1 a=np.array([-99,22,87,5,43,56,78,55,67,88,99,12,34])  
2 plt.hist(a)
```

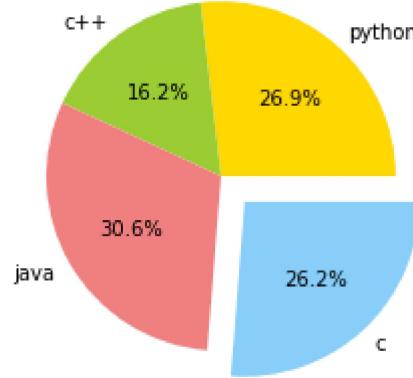
```
Out[28]: (array([1., 0., 0., 0., 0., 2., 2., 3., 2., 3.]),  
 array([-99. , -79.2, -59.4, -39.6, -19.8,  0. ,  19.8,  39.6,  59.4,  
        79.2,  99. ]),  
 <BarContainer object of 10 artists>)
```



pieplot

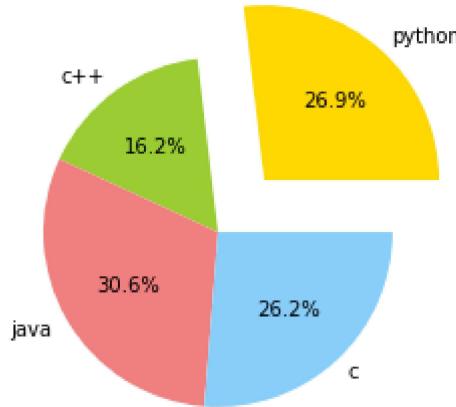
```
In [35]: 1 labels='python','c++','java','c'  
2 size=[215,130,245,210]  
3 colors=['gold','yellowgreen','lightcoral','lightskyblue']  
4 explode=(0,0,0,0.2)  
5 plt.pie(size,labels=labels,colors=colors,autopct="%1.1f%%",explode=explode)
```

```
Out[35]: ([<matplotlib.patches.Wedge at 0xb65c520>,  
<matplotlib.patches.Wedge at 0xb65cc10>,  
<matplotlib.patches.Wedge at 0xb667250>,  
<matplotlib.patches.Wedge at 0xb667850>],  
[Text(0.730677650908226, 0.8222591869132487, 'python'),  
Text(-0.64656382751384, 0.8899186574910392, 'c++'),  
Text(-0.94900779332915, -0.5562231640272972, 'java'),  
Text(0.8824410049439857, -0.9546192292183562, 'c')],  
[Text(0.39855144594994146, 0.44850501104359014, '26.9%'),  
Text(-0.35267117864391273, 0.485410176813294, '16.2%'),  
Text(-0.5176406145431727, -0.30339445310579843, '30.6%'),  
Text(0.5430406184270681, -0.5874579872112962, '26.2%')])
```



```
In [42]: 1 labels='python','c++','java','c'  
2 size=[215,130,245,210]  
3 colors=['gold','yellowgreen','lightcoral','lightskyblue']  
4 explode=(0.4,0,0,0)  
5 plt.pie(size,labels=labels,colors=colors,autopct='%1.1f%%',explode=explode)
```

```
Out[42]: ([<matplotlib.patches.Wedge at 0xa033eb0>,  
<matplotlib.patches.Wedge at 0xa2185e0>,  
<matplotlib.patches.Wedge at 0xa208ee0>,  
<matplotlib.patches.Wedge at 0xa305070>],  
[Text(0.9963786148748537, 1.1212625276089756, 'python'),  
Text(-0.64656382751384, 0.8899186574910392, 'c++'),  
Text(-0.94900779332915, -0.5562231640272972, 'java'),  
Text(0.7466808503372186, -0.8077547324155322, 'c')],  
[Text(0.6642524099165691, 0.747508351739317, '26.9%'),  
Text(-0.35267117864391273, 0.485410176813294, '16.2%'),  
Text(-0.5176406145431727, -0.30339445310579843, '30.6%'),  
Text(0.4072804638203011, -0.44059349040847207, '26.2%')])
```

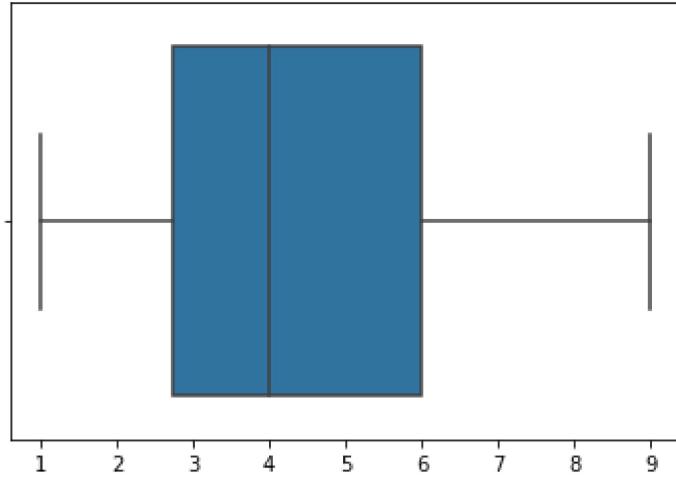


```
In [43]: 1 import seaborn as sns
```

```
In [44]: 1 x=[1,2,2,2,2,3,3,3,3,4,4,5,5,5,6,6,6,6,8,9]
          2 sns.boxplot(x)
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

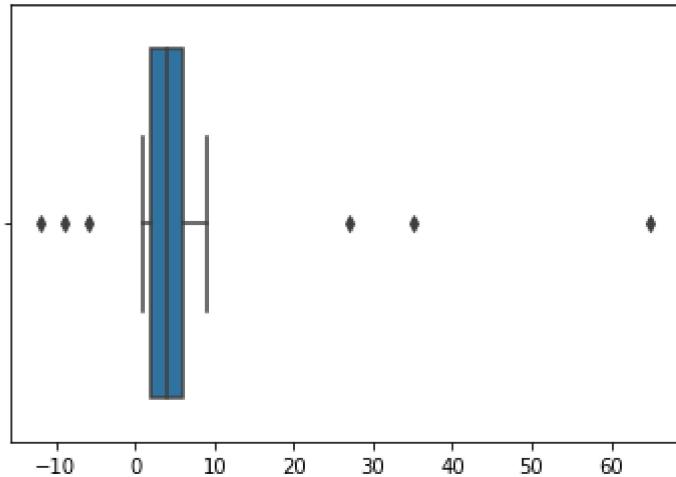
Out[44]: <AxesSubplot:>



```
In [48]: 1 x=[-12,-9,-6,1,2,2,2,2,3,3,3,3,4,4,5,5,5,6,6,6,6,8,9,27,35,65]
          2 sns.boxplot(x)
```

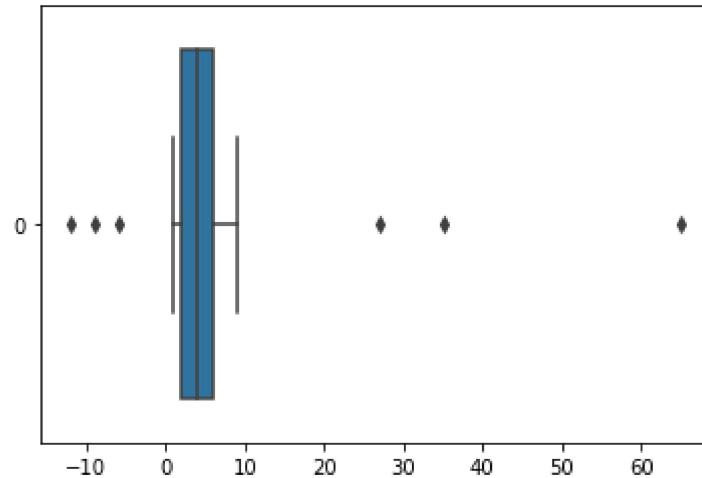
C:\Users\dell\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

Out[48]: <AxesSubplot:>



```
In [51]: 1 x=[-12,-9,-6,1,2,2,2,2,3,3,3,3,4,4,5,5,5,6,6,6,6,8,9,27,35,65]
          2 sns.boxplot(data=x,orient='h')
```

Out[51]: <AxesSubplot:>



seaborn

```
In [1]: 1 #distribution plots
      2 #distplot
      3 #joinplot
      4 #pairplot
      5
```

```
In [6]: 1 import seaborn as sns
      2 df=sns.load_dataset('tips')
      3 df.head()
```

Out[6]:

	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 [2]: 1 #correlation
```

```
In [7]: 1 df.corr()
```

```
-----  
ValueError                                                 Traceback (most recent call last)  
Cell In[7], line 1  
----> 1 df.corr()  
  
File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:10054, in DataFrame.corr(self, method, min_periods, numeric_only)  
10052     cols = data.columns  
10053     idx = cols.copy()  
> 10054     mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)  
10056     if method == "pearson":  
10057         correl = libalgos.nancorr(mat, minp=min_periods)  
  
File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:1838, in DataFrame.to_numpy(self, dtype, copy, na_value)  
1836     if dtype is not None:  
1837         dtype = np.dtype(dtype)  
-> 1838     result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)  
1839     if result.dtype is not dtype:  
1840         result = np.array(result, dtype=dtype, copy=False)  
  
File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1732, in BlockManager.as_array(self, dtype, copy, na_value)  
1730         arr.flags.writeable = False  
1731     else:  
-> 1732         arr = self._interleave(dtype=dtype, na_value=na_value)  
1733         # The underlying data was copied within _interleave, so no need  
1734         # to further copy if copy=True or setting na_value  
1736     if na_value is not lib.no_default:  
  
File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1788, in BlockManager._interleave(self, dtype, na_value)  
1782     rl = blk.mgr_locs  
1783     if blk.is_extension:  
1784         # Avoid implicit conversion of extension blocks to object  
1785  
1786         # error: Item "ndarray" of "Union[ndarray, ExtensionArray]" has n  
o  
1787         # attribute "to_numpy"  
-> 1788         arr = blk.values.to_numpy( # type: ignore[union-attr]  
1789             dtype=dtype,  
1790             na_value=na_value,  
1791         )  
1792     else:  
1793         arr = blk.get_values(dtype)  
  
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\base.py:485, in ExtensionArray.to_numpy(self, dtype, copy, na_value)  
456     def to_numpy(  
457         self,  
458         dtype: npt.DTypeLike | None = None,  
459         copy: bool = False,  
460         na_value: object = lib.no_default,  
461     ) -> np.ndarray:  
462         """  
463             Convert to a NumPy ndarray.
```

```
464
(...)  
483     numpy.ndarray  
484     """  
--> 485     result = np.asarray(self, dtype=dtype)  
486     if copy or na_value is not lib.no_default:  
487         result = result.copy()  
  
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\_mixins.py:86, in ravel  
_compat.<locals>.method(self, *args, **kwargs)  
    83 @wraps(meth)  
    84 def method(self, *args, **kwargs):  
    85     if self.ndim == 1:  
---> 86         return meth(self, *args, **kwargs)  
    88     flags = self._ndarray.flags  
    89     flat = self.ravel("K")  
  
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\categorical.py:1344, in  
Categorical.__array__(self, dtype)  
1342 ret = take_nd(self.categories._values, self._codes)  
1343 if dtype and not is_dtype_equal(dtype, self.categories.dtype):  
-> 1344     return np.asarray(ret, dtype)  
1345 # When we're a Categorical[ExtensionArray], like Interval,  
1346 # we need to ensure __array__ gets all the way to an  
1347 # ndarray.  
1348 return np.asarray(ret)  
  
ValueError: could not convert string to float: 'No'
```

```
In [5]: 1 sns.heatmap(df.corr())
```

```

-----
ValueError                                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 sns.heatmap(df.corr())

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:10054, in DataFrame.corr(self, method, min_periods, numeric_only)
10052     cols = data.columns
10053     idx = cols.copy()
> 10054     mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
10056     if method == "pearson":
10057         correl = libalgos.nancorr(mat, minp=min_periods)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:1838, in DataFrame.to_numpy(self, dtype, copy, na_value)
1836     if dtype is not None:
1837         dtype = np.dtype(dtype)
-> 1838     result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
1839     if result.dtype is not dtype:
1840         result = np.array(result, dtype=dtype, copy=False)

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1732, in BlockManager.as_array(self, dtype, copy, na_value)
1730         arr.flags.writeable = False
1731     else:
-> 1732         arr = self._interleave(dtype=dtype, na_value=na_value)
1733         # The underlying data was copied within _interleave, so no need
1734         # to further copy if copy=True or setting na_value
1736     if na_value is not lib.no_default:

File ~\anaconda3\Lib\site-packages\pandas\core\internals\managers.py:1788, in BlockManager._interleave(self, dtype, na_value)
1782     rl = blk.mgr_locs
1783     if blk.is_extension:
1784         # Avoid implicit conversion of extension blocks to object
1785
1786         # error: Item "ndarray" of "Union[ndarray, ExtensionArray]" has n
o
1787         # attribute "to_numpy"
-> 1788         arr = blk.values.to_numpy( # type: ignore[union-attr]
1789             dtype=dtype,
1790             na_value=na_value,
1791         )
1792     else:
1793         arr = blk.get_values(dtype)

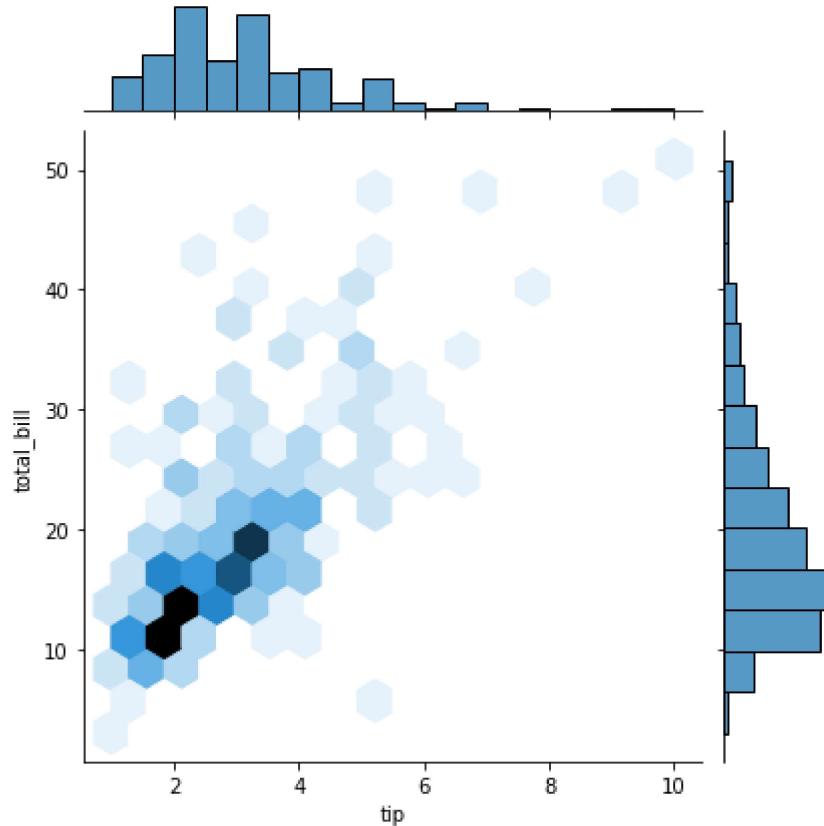
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\base.py:485, in ExtensionArray.to_numpy(self, dtype, copy, na_value)
456     def to_numpy(
457         self,
458         dtype: npt.DTypeLike | None = None,
459         copy: bool = False,
460         na_value: object = lib.no_default,
461     ) -> np.ndarray:
462         """
463         Convert to a NumPy ndarray.

```

```
464
(...)  
483     numpy.ndarray  
484     """  
---> 485     result = np.asarray(self, dtype=dtype)  
486     if copy or na_value is not lib.no_default:  
487         result = result.copy()  
  
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\_mixins.py:86, in ravel  
_compat.<locals>.method(self, *args, **kwargs)  
    83 @wraps(meth)  
    84 def method(self, *args, **kwargs):  
    85     if self.ndim == 1:  
---> 86         return meth(self, *args, **kwargs)  
    88     flags = self._ndarray.flags  
    89     flat = self.ravel("K")  
  
File ~\anaconda3\Lib\site-packages\pandas\core\arrays\categorical.py:1344, in  
Categorical.__array__(self, dtype)  
1342 ret = take_nd(self.categories._values, self._codes)  
1343 if dtype and not is_dtype_equal(dtype, self.categories.dtype):  
-> 1344     return np.asarray(ret, dtype)  
1345 # When we're a Categorical[ExtensionArray], like Interval,  
1346 # we need to ensure __array__ gets all the way to an  
1347 # ndarray.  
1348 return np.asarray(ret)  
  
ValueError: could not convert string to float: 'No'
```

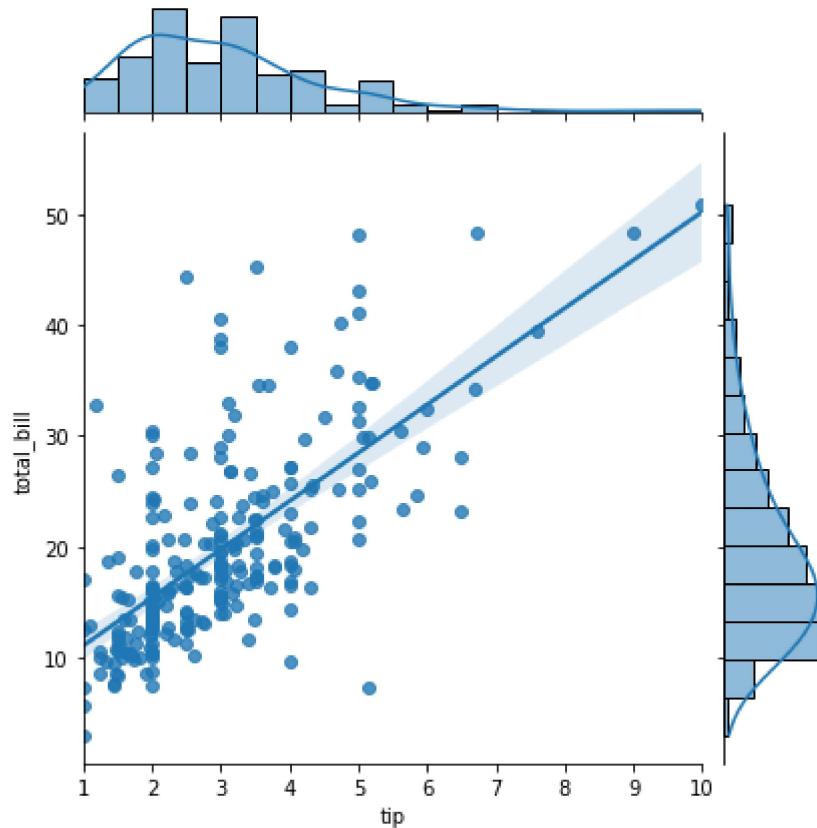
```
In [9]: 1 sns.jointplot(x='tip',y='total_bill',data=df,kind='hex')
```

```
Out[9]: <seaborn.axisgrid.JointGrid at 0xd71c340>
```



```
In [11]: 1 sns.jointplot(x='tip',y='total_bill',data=df,kind='reg')
```

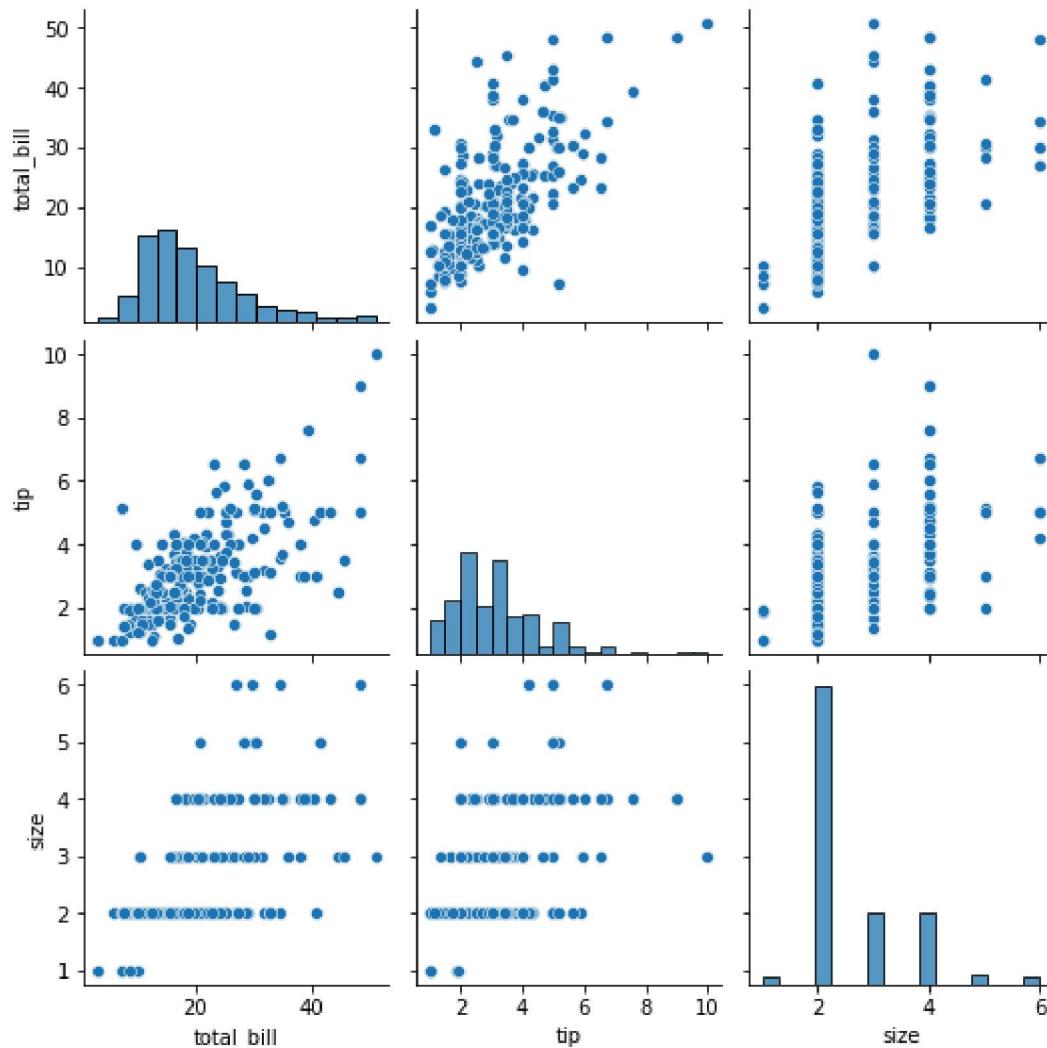
```
Out[11]: <seaborn.axisgrid.JointGrid at 0xeae1490>
```



```
In [12]: 1 #pairplot
```

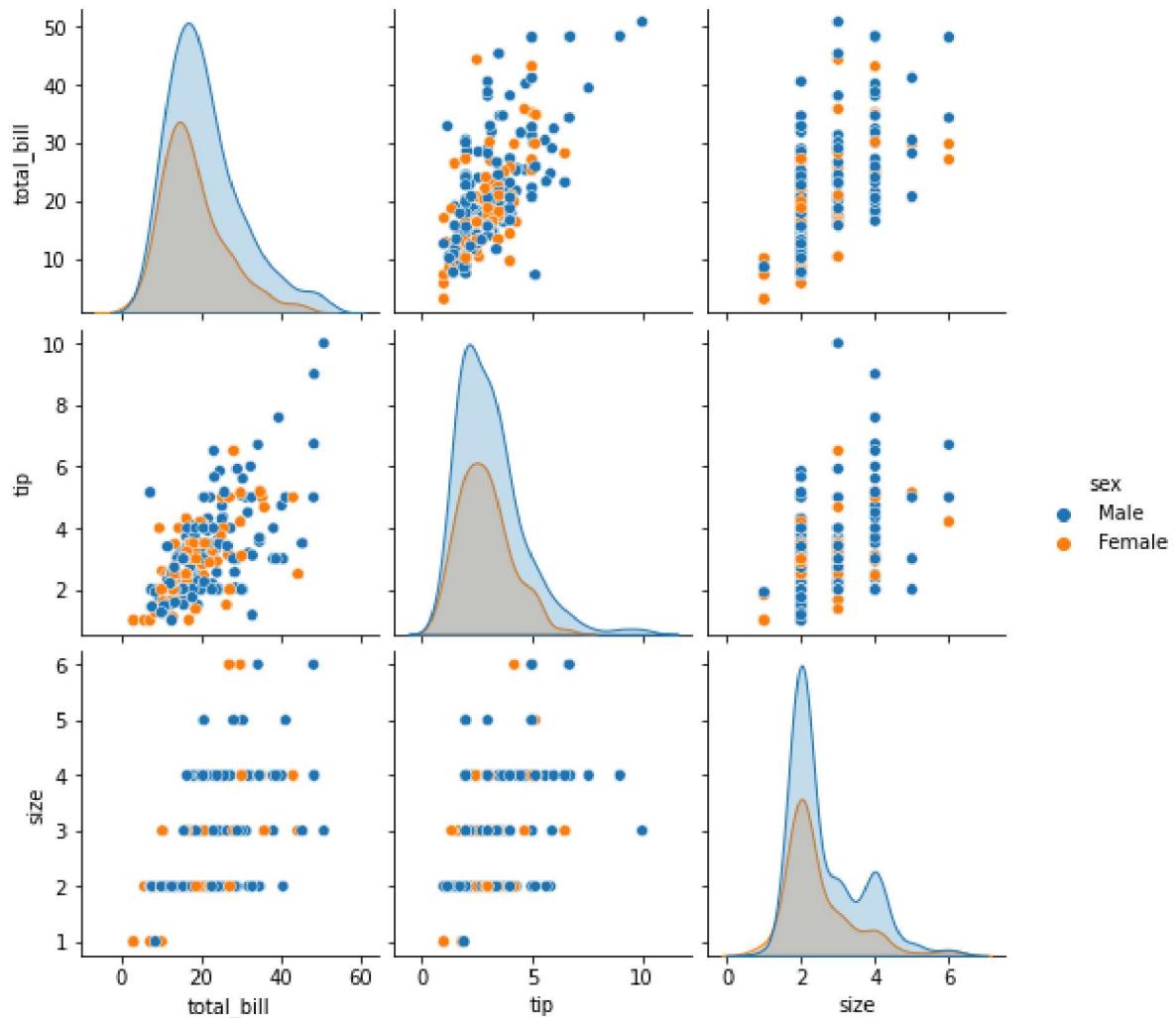
```
In [13]: 1 sns.pairplot(df)
```

```
Out[13]: <seaborn.axisgrid.PairGrid at 0xeb80490>
```



```
In [14]: 1 sns.pairplot(df,hue='sex')
```

```
Out[14]: <seaborn.axisgrid.PairGrid at 0xd9b83d0>
```



```
In [15]: 1 #distplot
```

In [16]: 1 df

Out[16]:

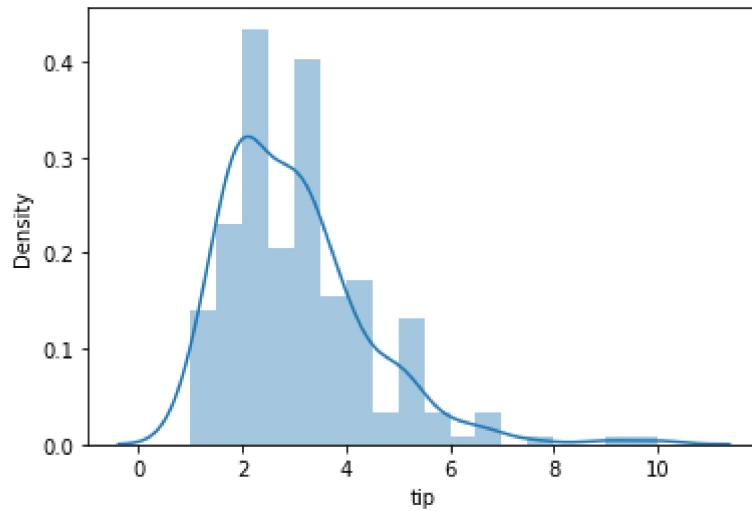
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [17]: 1 sns.distplot(df['tip'])

C:\Users\dell\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
 warnings.warn(msg, FutureWarning)

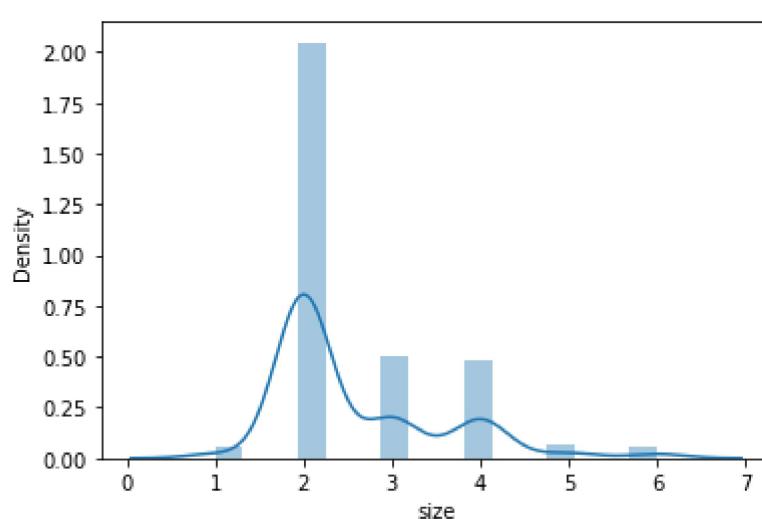
Out[17]: <AxesSubplot:xlabel='tip', ylabel='Density'>



```
In [20]: 1 sns.distplot(df['size'])
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

Out[20]: <AxesSubplot:xlabel='size', ylabel='Density'>

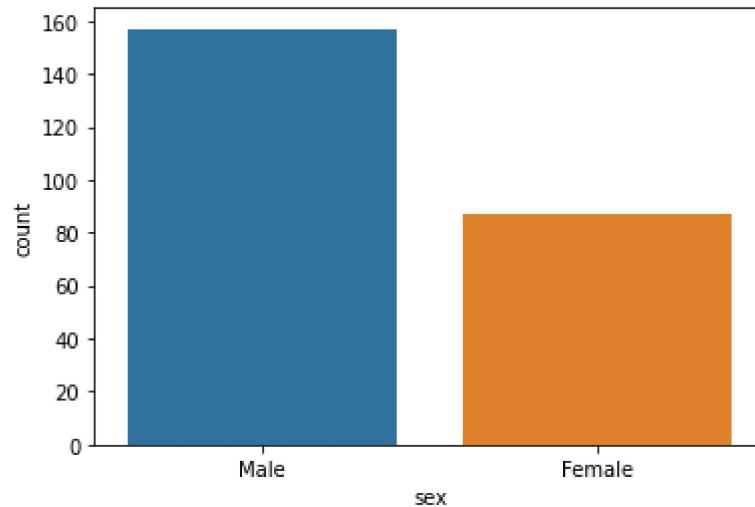


```
In [22]: 1 #count plot
```

```
2  
3 sns.countplot('sex', data=df)
```

C:\Users\dell\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

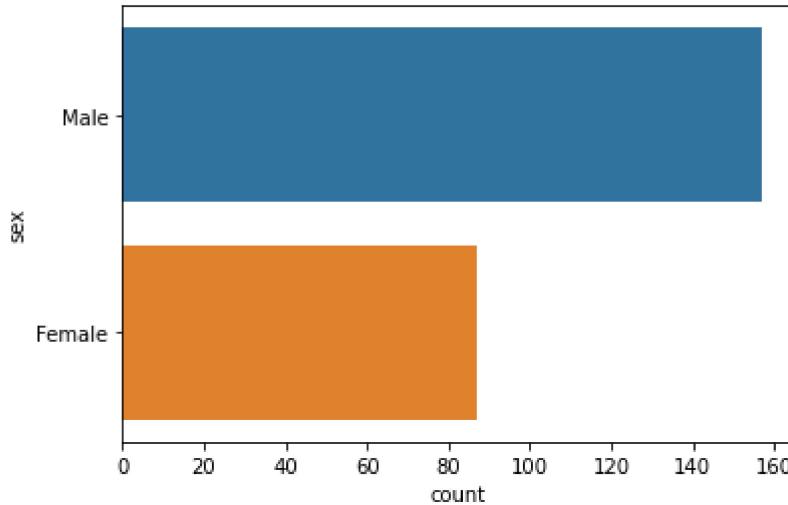
Out[22]: <AxesSubplot:xlabel='sex', ylabel='count'>



In [23]:

```
1
2 sns.countplot(y='sex', data=df)
```

Out[23]: <AxesSubplot:xlabel='count', ylabel='sex'>



In []:

```
1
```