infosys

November 3, 2025

EXAMPLE

[ ]:
```
print('hi')
```

```
hi
```

1 NUMPY

NumPy is a fundamental library for numerical computing in Python. It provides support for arrays and matrices, along with a large collection of mathematical functions to operate on these arrays.

[ ]:
```
#IMPORTING NUMPY LIBRARY
import numpy as np
```

[ ]:
```
#single dimensional array
n1=np.array([1,2,3,4,5])
n1
```

[ ]:
```
array([1, 2, 3, 4, 5])
```

[ ]:
```
#multidimensional array
n2=np.array([[1,2,3],[4,5,6]])
n2
```

[ ]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

[ ]:
```
#index value
n1[0]
```

[ ]:
```
np.int64(1)
```

[ ]:
```
n1[1]
```

[ ]:
```
np.int64(2)
```

[ ]:
```
n1=np.array([[1,2,3],[4,5,6],[7,8,9]])
n1[:,1]
```

```
[ ]: array([2, 5, 8])
```

```
[ ]: n1[:,2]
```

```
[ ]: array([3, 6, 9])
```

```
[ ]: #Initializing Numpy array with zeros
     n1=np.zeros((1,2))
     n1
```

```
[ ]: array([[0., 0.]])
```

```
[ ]: #multidimensional array
     n2=np.zeros((4,4))
     n2
```

```
[ ]: array([[0., 0., 0., 0.],
            [0., 0., 0., 0.],
            [0., 0., 0., 0.],
            [0., 0., 0., 0.]])
```

```
[ ]: #initializing numpy array with same numbers
     n1=np.full((2,2),10)
     n1
```

```
[ ]: array([[10, 10],
            [10, 10]])
```

```
[ ]: #initializing numpy array
     n1=np.arange(10,20)
     n1
```

```
[ ]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
[ ]: #with difference
     n1=np.arange(10,40,2)
     n1
```

```
[ ]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38])
```

```
[ ]: #random integers
     n1=np.random.randint(1,45,3)
     n1
```

```
[ ]: array([19, 23, 14])
```

[ ]:
```
#shape
n1=np.array([[1,2,3],[4,5,6]])
n1.shape
```

[ ]:
```
(2, 3)
```

[ ]:
```
#vstack() stacks arrays vertically(row-wise)
n1=np.array([10,20,30])
n2=np.array([40,50,60])
np.vstack((n1,n2))
```

[ ]:
```
array([[10, 20, 30],
       [40, 50, 60]])
```

[ ]:
```
#Hstack() stacks arrays horizontally ( column-wise)
n1= np.array([10,20,30])
n2=np.array([40,50,60])
np.hstack((n1,n2))
```

[ ]:
```
array([10, 20, 30, 40, 50, 60])
```

[ ]:
```
#COLUMN_STACK() stacks 1-D arrays as columns into a 2D-array
n1=np.array([10,20,30])
n2=np.array([40,50,60])
np.column_stack((n1,n2))
```

[ ]:
```
array([[10, 40],
       [20, 50],
       [30, 60]])
```

[ ]:
```
#intersection
np.intersect1d(n1,n2)
```

[ ]:
```
array([], dtype=int64)
```

[ ]:
```
#intersection
n3=np.array([20,40,60])
np.intersect1d(n1,n3)
```

[ ]:
```
array([20])
```

[ ]:
```
#difference
np.setdiff1d(n1,n2)
```

[ ]:
```
array([10, 20, 30])
```

[ ]:
```
np.setdiff1d(n2,n1)
```

```
[ ]: array([40, 50, 60])
```

```
[ ]: #total sum
     np.sum([n1,n2])
```

```
[ ]: np.int64(210)
```

```
[ ]: #sum of respective coordinate axis
     np.sum([n1,n2],axis=0)
```

```
[ ]: array([50, 70, 90])
```

```
[ ]: #addition
     n1=n1+1
     n1
```

```
[ ]: array([11, 21, 31])
```

```
[ ]: #multiplication
     n1=np.array([[1,2,3],[4,5,6],[7,8,9]])
     n2=np.array([[1,2,3],[3,4,5],[5,6,7]])
     n1.dot(n2)
```

```
[ ]: array([[ 22,  28,  34],
            [ 49,  64,  79],
            [ 76, 100, 124]])
```

```
[ ]: n2.dot(n1.T)
```

```
[ ]: array([[ 14,  32,  50],
            [ 26,  62,  98],
            [ 38,  92, 146]])
```

```
[ ]: n1=n1*2
     n1
```

```
[ ]: array([[ 2,  4,  6],
            [ 8, 10, 12],
            [14, 16, 18]])
```

```
[ ]: #division
     n1=n1/2
     n1
```

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

```
[ ]: #mean
     np.mean(n1)
```

```
[ ]: np.float64(5.0)
```

```
[ ]: #median
     np.median(n1)
```

```
[ ]: np.float64(5.0)
```

```
[ ]: #standard deviation
     np.std(n1)
```

```
[ ]: np.float64(2.581988897471611)
```

```
[ ]: #transpose()
     n1.transpose()
```

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

2 PANDA

```
[ ]: #importing pandas library
     import pandas as pd
```

```
[ ]: #series object is one-dimensional labeled array
     s1=pd.Series([10,20,30,40,50])
     s1
```

```
[ ]: 0    10
     1    20
     2    30
     3    40
     4    50
     dtype: int64
```

```
[ ]: #userdefined index
     s1=pd.Series([10,20,30],index=['x','y','z'])
     s1
```

```
[ ]: x    10
     y    20
     z    30
     dtype: int64
```

```
[ ]: #dictionary
     pd.Series({'a':10,'b':20,'c':30})
```

```
[ ]: a    10
     b    20
     c    30
     dtype: int64
```

```
[ ]: #missed data
     pd.Series({'a':10,'b':20,'c':30},index=['b','a','g'])
```

```
[ ]: b    20.0
     a    10.0
     g     NaN
     dtype: float64
```

```
[ ]: #extracting a single element
     s1=pd.Series([10,20,30,40,50])
     s1[3]
```

```
[ ]: np.int64(40)
```

```
[ ]: #extracting a sequence of elements
     s1=pd.Series([10,20,30,40,50])
     s1[:4]
```

```
[ ]: 0    10
     1    20
     2    30
     3    40
     dtype: int64
```

```
[ ]: #extracting elements from back
     s2=pd.Series([10,20,30,40,50])
     s2[-3:]
```

```
[ ]: 2    30
     3    40
     4    50
     dtype: int64
```

```
[ ]: s1+s2
```

```
[ ]: 0     20
     1     40
     2     60
     3     80
```

```
4    100
dtype: int64
```

```
[ ]: s1+5
```

```
[ ]: 0    15
     1    25
     2    35
     3    45
     4    55
     dtype: int64
```

```
[ ]: #creating dataframes
     pd.DataFrame({"Name":['A','B','C'],'Marks':[90,75,82]})
```

```
[ ]:   Name  Marks
     0    A     90
     1    B     75
     2    C     82
```

```
[ ]: #defining through variable
     df=pd.DataFrame({"Name":['A','B','C'],'Marks':[90,75,82]})
     df
```

```
[ ]:   Name  Marks
     0    A     90
     1    B     75
     2    C     82
```

```
[ ]: type(df)
```

```
[ ]: pandas.core.frame.DataFrame
```

3 *IRIS DATASET*

```
[ ]: from google.colab import files
     uploaded=files.upload()
```

```
<IPython.core.display.HTML object>

Saving IRIS.csv to IRIS (1).csv
```

```
[ ]: df=pd.read_csv('IRIS.csv')
     print(df.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width      species
0           5.1          3.5           1.4          0.2  Iris-setosa
```

```
1           4.9         3.0          1.4          0.2  Iris-setosa
2           4.7         3.2          1.3          0.2  Iris-setosa
3           4.6         3.1          1.5          0.2  Iris-setosa
4           5.0         3.6          1.4          0.2  Iris-setosa
```

[ ]:

```
#head()-displays first n rows by default 5
df.head(10)
```

[ ]:

```
   sepal_length  sepal_width  petal_length  petal_width      species
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa
5           5.4          3.9           1.7          0.4  Iris-setosa
6           4.6          3.4           1.4          0.3  Iris-setosa
7           5.0          3.4           1.5          0.2  Iris-setosa
8           4.4          2.9           1.4          0.2  Iris-setosa
9           4.9          3.1           1.5          0.1  Iris-setosa
```

[ ]:

```
#displays last n rows by default 5
df.tail()
```

[ ]:

```
     sepal_length  sepal_width  petal_length  petal_width         species
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica
```

[ ]:

```
df.tail(10)
```

[ ]:

```
     sepal_length  sepal_width  petal_length  petal_width         species
140           6.7          3.1           5.6          2.4  Iris-virginica
141           6.9          3.1           5.1          2.3  Iris-virginica
142           5.8          2.7           5.1          1.9  Iris-virginica
143           6.8          3.2           5.9          2.3  Iris-virginica
144           6.7          3.3           5.7          2.5  Iris-virginica
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica
```

[ ]:

```
df.shape
```

[ ]:

```
(150, 5)
```

```
[ ]: #generates descriptiv statistics of a dataframe
     df.describe()
```

```
[ ]:        sepal_length  sepal_width  petal_length  petal_width
     count    150.000000   150.000000    150.000000   150.000000
     mean       5.843333     3.054000      3.758667     1.198667
     std        0.828066     0.433594      1.764420     0.763161
     min        4.300000     2.000000      1.000000     0.100000
     25%        5.100000     2.800000      1.600000     0.300000
     50%        5.800000     3.000000      4.350000     1.300000
     75%        6.400000     3.300000      5.100000     1.800000
     max        7.900000     4.400000      6.900000     2.500000
```

```
[ ]: #integer-location based indexing/selection by position
     df.iloc[0:4,0:2]
```

```
[ ]:    sepal_length  sepal_width
     0           5.1          3.5
     1           4.9          3.0
     2           4.7          3.2
     3           4.6          3.1
```

```
[ ]: df.iloc[0:5,3:5]
```

```
[ ]:    petal_width      species
     0          0.2  Iris-setosa
     1          0.2  Iris-setosa
     2          0.2  Iris-setosa
     3          0.2  Iris-setosa
     4          0.2  Iris-setosa
```

```
[ ]: df.iloc[10:20,2:5]
```

```
[ ]:     petal_length  petal_width      species
     10           1.5          0.2  Iris-setosa
     11           1.6          0.2  Iris-setosa
     12           1.4          0.1  Iris-setosa
     13           1.1          0.1  Iris-setosa
     14           1.2          0.2  Iris-setosa
     15           1.5          0.4  Iris-setosa
     16           1.3          0.4  Iris-setosa
     17           1.4          0.3  Iris-setosa
     18           1.7          0.3  Iris-setosa
     19           1.5          0.3  Iris-setosa
```

```
[ ]: #removes specified rows or columns
     df.drop('sepal_length',axis=1)
```

```
[ ]:      sepal_width  petal_length  petal_width          species
     0            3.5           1.4          0.2      Iris-setosa
     1            3.0           1.4          0.2      Iris-setosa
     2            3.2           1.3          0.2      Iris-setosa
     3            3.1           1.5          0.2      Iris-setosa
     4            3.6           1.4          0.2      Iris-setosa
     ..           …           …          …             …
     145          3.0           5.2          2.3   Iris-virginica
     146          2.5           5.0          1.9   Iris-virginica
     147          3.0           5.2          2.0   Iris-virginica
     148          3.4           5.4          2.3   Iris-virginica
     149          3.0           5.1          1.8   Iris-virginica

     [150 rows x 4 columns]
```

```
[ ]: #label-based indexing/selection by label
     df.loc[3:6,('sepal_length','petal_length')]
```

```
[ ]:    sepal_length  petal_length
     3           4.6           1.5
     4           5.0           1.4
     5           5.4           1.7
     6           4.6           1.4
```

```
[ ]: df.drop([1,2,3],axis=0)
```

```
[ ]:      sepal_length  sepal_width  petal_length  petal_width          species
     0             5.1          3.5           1.4          0.2      Iris-setosa
     4             5.0          3.6           1.4          0.2      Iris-setosa
     5             5.4          3.9           1.7          0.4      Iris-setosa
     6             4.6          3.4           1.4          0.3      Iris-setosa
     7             5.0          3.4           1.5          0.2      Iris-setosa
     ..            …          …           …          …             …
     145           6.7          3.0           5.2          2.3   Iris-virginica
     146           6.3          2.5           5.0          1.9   Iris-virginica
     147           6.5          3.0           5.2          2.0   Iris-virginica
     148           6.2          3.4           5.4          2.3   Iris-virginica
     149           5.9          3.0           5.1          1.8   Iris-virginica

     [147 rows x 5 columns]
```

```
[ ]: df.mean(numeric_only=True)
```

```
[ ]: sepal_length    5.843333
     sepal_width     3.054000
     petal_length    3.758667
     petal_width     1.198667
```

```
dtype: float64
```

```
[ ]: df.median(numeric_only=True)
```

```
[ ]: sepal_length    5.80
     sepal_width     3.00
     petal_length    4.35
     petal_width     1.30
     dtype: float64
```

```
[ ]: df.min()
```

```
[ ]: sepal_length            4.3
     sepal_width             2.0
     petal_length            1.0
     petal_width             0.1
     species         Iris-setosa
     dtype: object
```

```
[ ]: df.max()
```

```
[ ]: sepal_length               7.9
     sepal_width                4.4
     petal_length               6.9
     petal_width                2.5
     species         Iris-virginica
     dtype: object
```

```
[ ]: print(df.loc[0:4,['sepal_length','petal_length']])
     df.drop('petal_width',axis=1,inplace=True)
     df.drop([0,1],axis=0,inplace=True)
     num=df.select_dtypes(include=['number'])
     print("Mean:\n",num.mean())
     print("Median:\n",num.median())
     print("Min:\n",num.min())
     print("Max:\n",num.max())
```

```
   sepal_length  petal_length
0           5.1           1.4
1           4.9           1.4
2           4.7           1.3
3           4.6           1.5
4           5.0           1.4
Mean:
 sepal_length    5.854730
sepal_width     3.051351
petal_length    3.790541
```

```
dtype: float64
Median:
 sepal_length    5.8
sepal_width     3.0
petal_length    4.4
dtype: float64
Min:
 sepal_length    4.3
sepal_width     2.0
petal_length    1.0
dtype: float64
Max:
 sepal_length    7.9
sepal_width     4.4
petal_length    6.9
dtype: float64
```

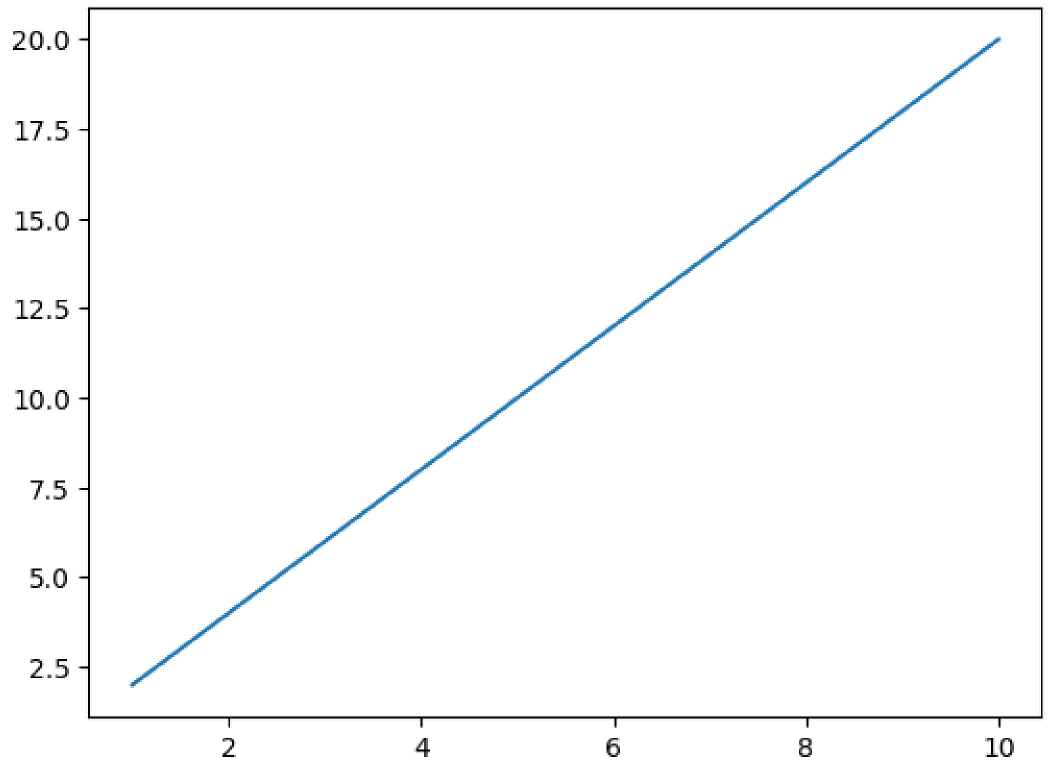
4 MATPLOTLIB

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

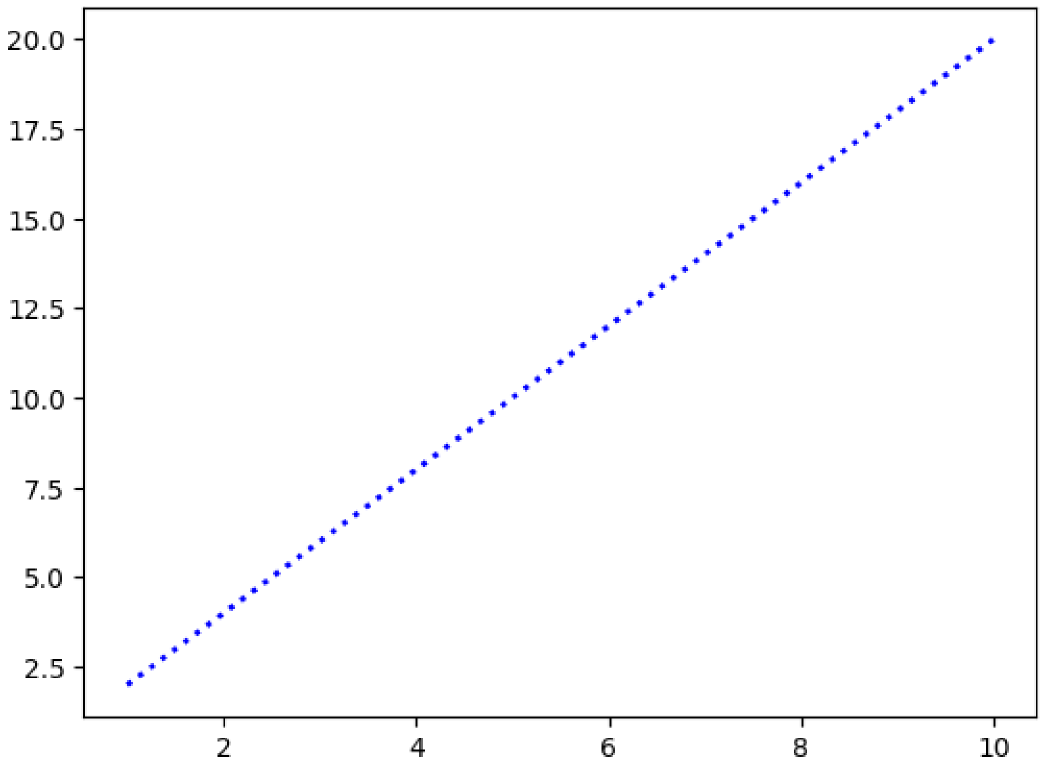
[ ]:
```
x=np.arange(1,11)
y=2*x
print('x:',x)
print('y:',y)
```

```
x: [ 1  2  3  4  5  6  7  8  9 10]
y: [ 2  4  6  8 10 12 14 16 18 20]
```

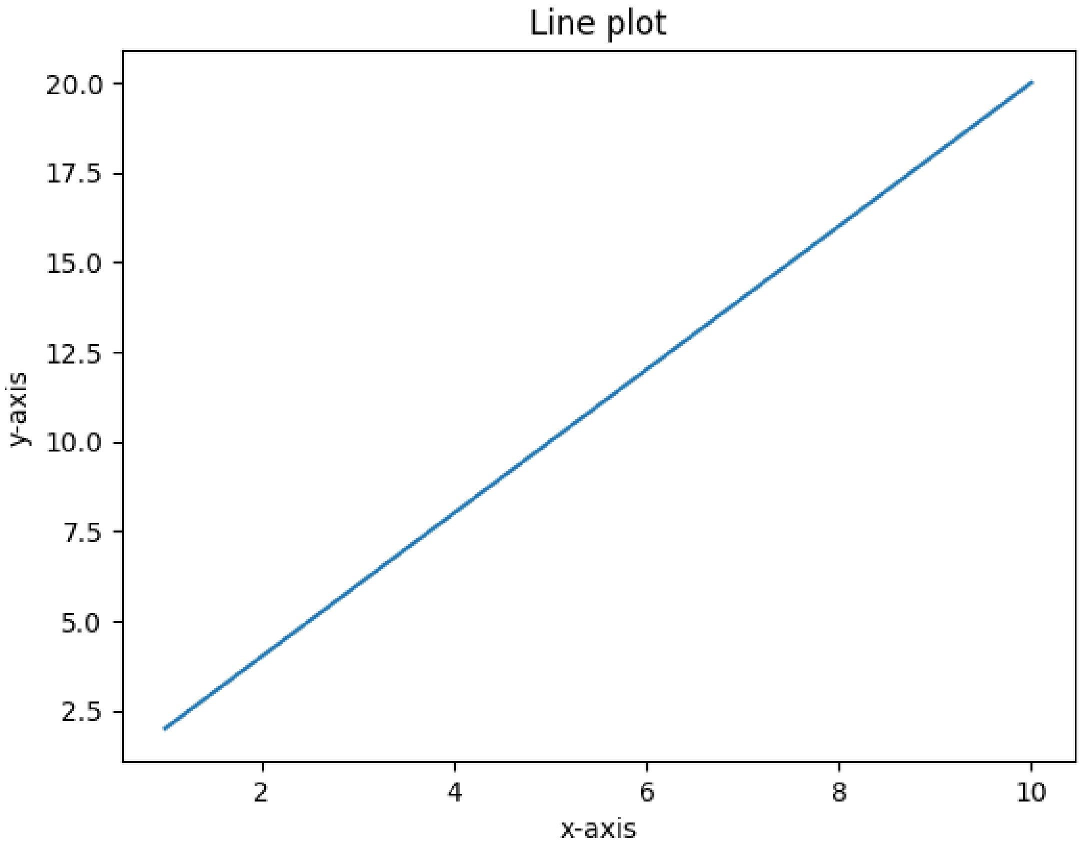
[ ]:
```
plt.plot(x,y)
plt.show()
```



[ ]:
```
plt.plot(x,y,color='blue',linestyle=':',linewidth=2)
plt.show()
```

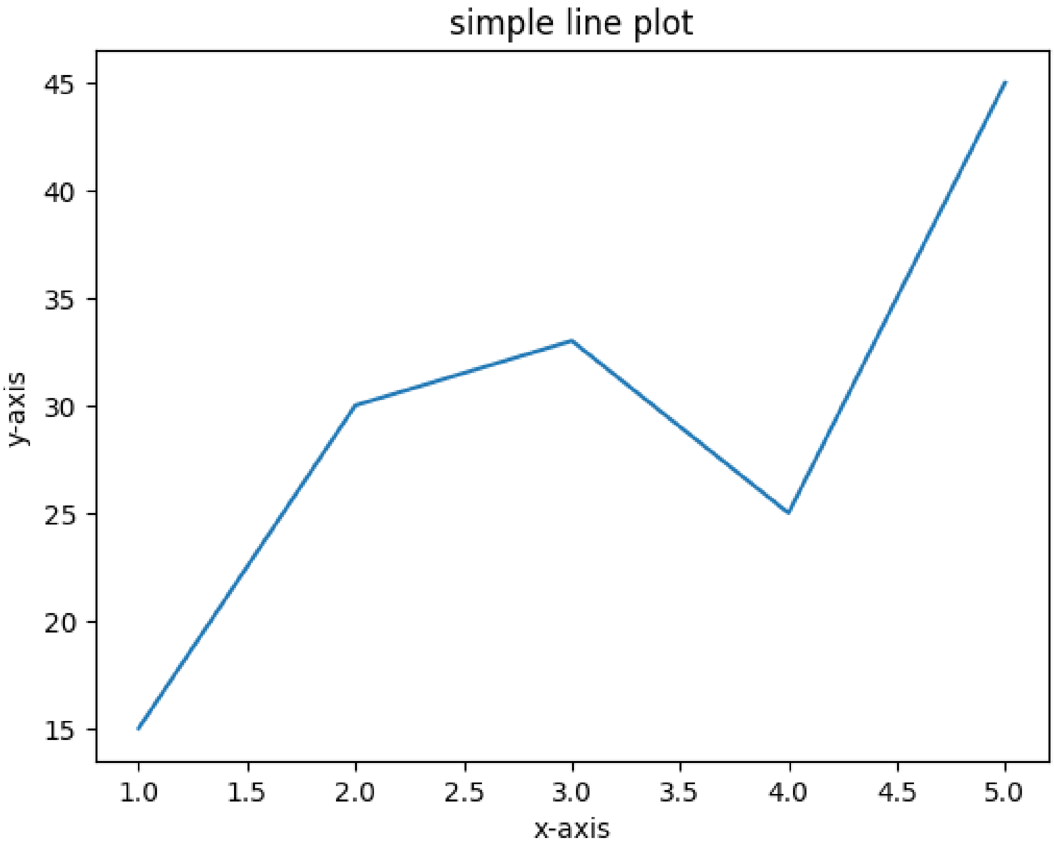


```
[ ]: plt.plot(x,y)
     plt.title("Line plot")
     plt.xlabel("x-axis")
     plt.ylabel("y-axis")
     plt.show()
```



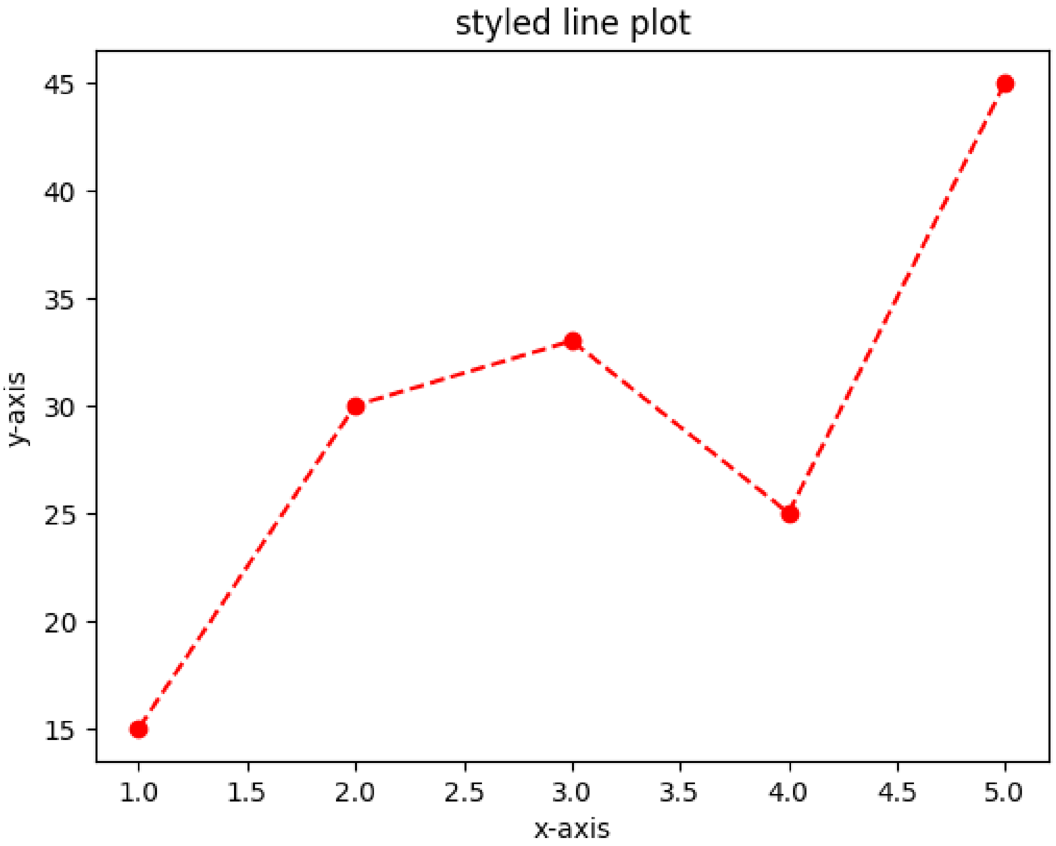
[ ]:

```
x=[1,2,3,4,5]
y=[15,30,33,25,45]
plt.plot(x,y)
plt.title('simple line plot')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.show()
```



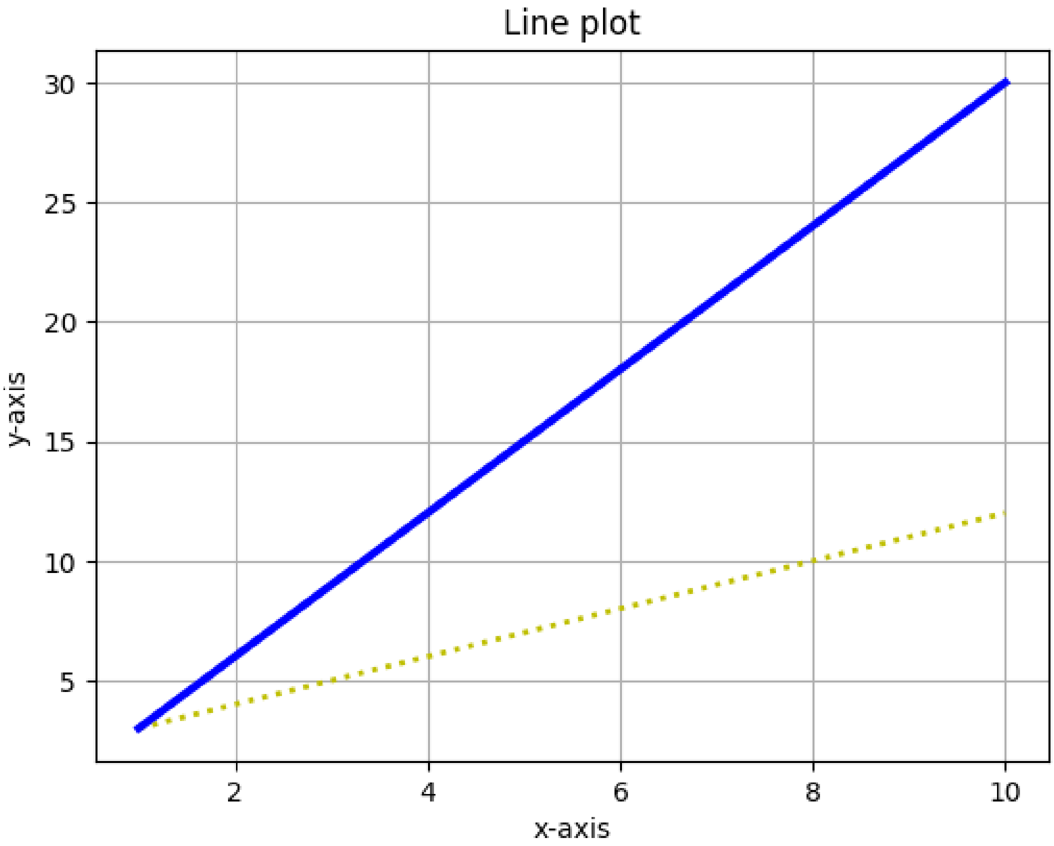
[ ]:

```
plt.plot(x,y,color='red',linestyle='--',marker='o')
plt.title('styled line plot')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.show()
```



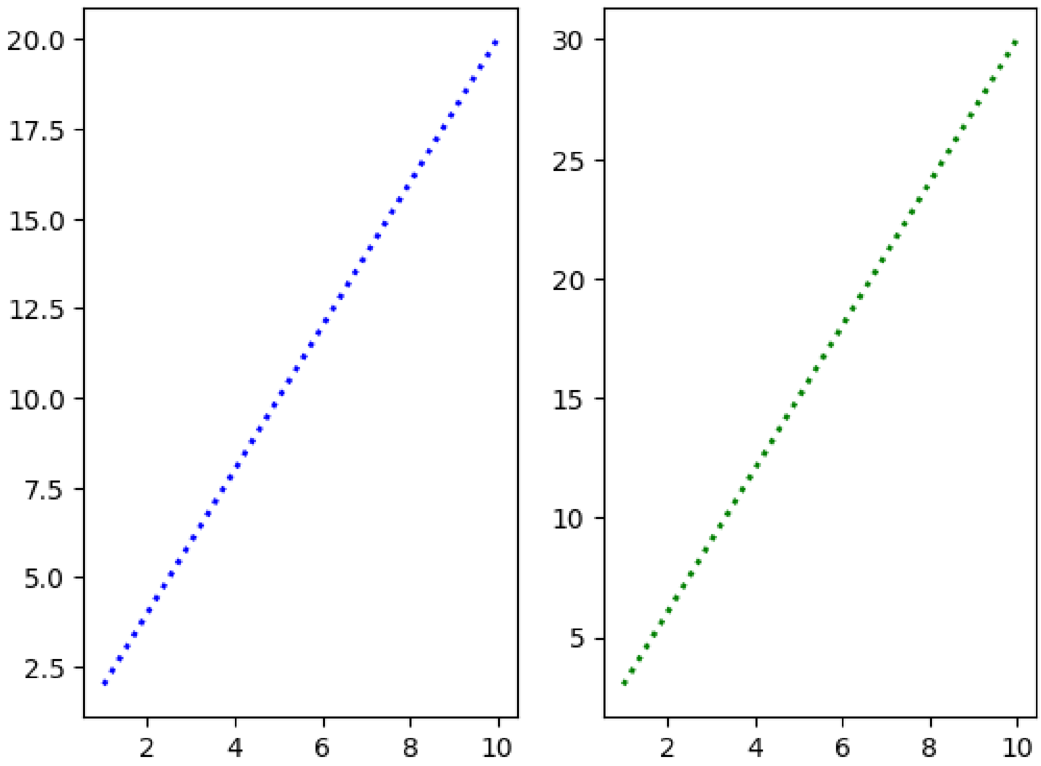
[ ]:

```
x=np.arange(1,11)
y1=2+x
y2=3*x
plt.plot(x,y1,color='y',linestyle=':',linewidth=2)
plt.plot(x,y2,color='b',linestyle='-',linewidth=3)
plt.title("Line plot")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.grid(True)
plt.show()
```



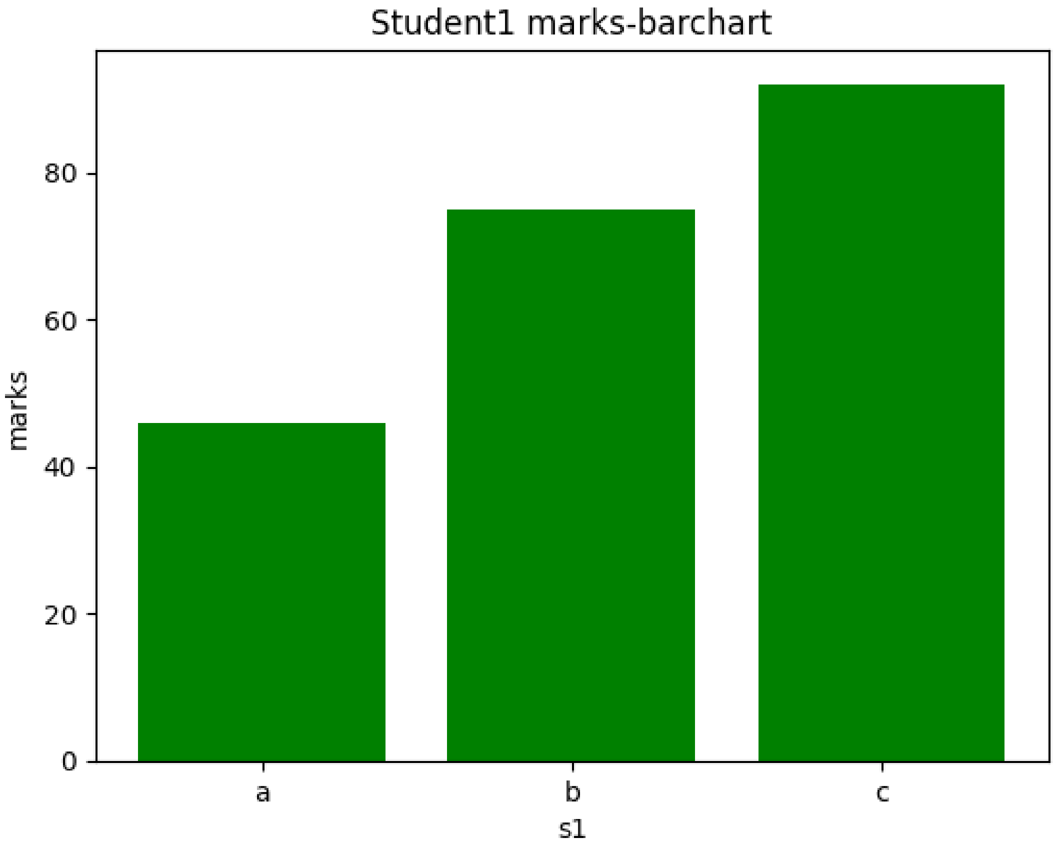
[ ]:

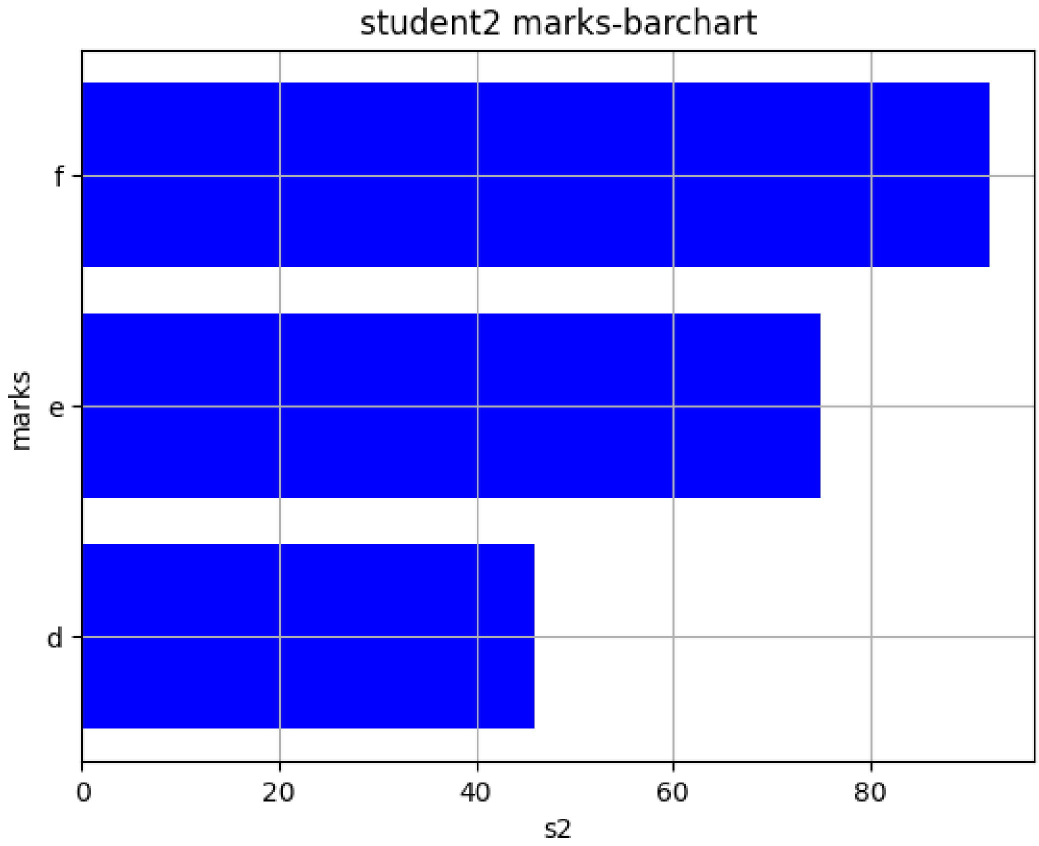
```
#adds a subplot to a current figure
x=np.arange(1,11)
y1=2*x
y2=3*x
plt.subplot(1,2,1)
plt.plot(x,y1,color='b',linestyle=':',linewidth=2)
plt.subplot(1,2,2)
plt.plot(x,y2,color='g',linestyle=':',linewidth=2)
plt.show()
```



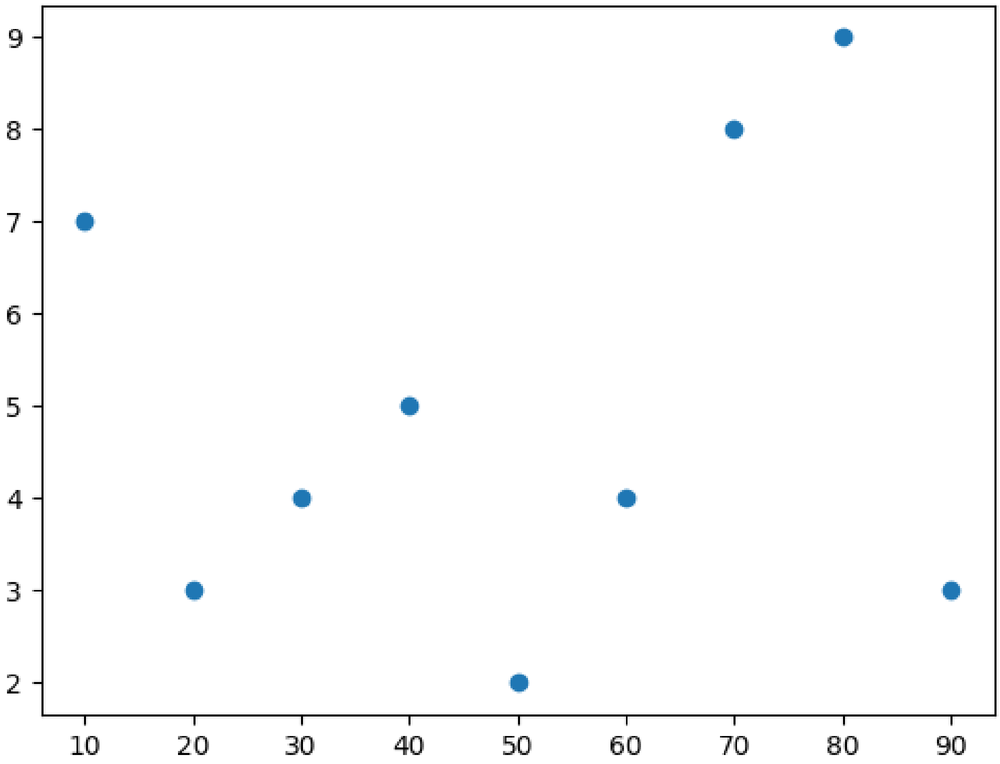
[ ]:

```
s1=['a','b','c']
s2=['d','e','f']
marks=[46,75,92]
plt.bar(s1,marks,color='g')
plt.title('Student1 marks-barchart')
plt.xlabel('s1')
plt.ylabel('marks')
plt.show()
plt.barh(s2,marks,color='b')
plt.title('student2 marks-barchart')
plt.xlabel('s2')
plt.ylabel('marks')
plt.grid(True)
plt.show()
```


Student1 marks-barchart
marks
0
20
40
60
80
a
b
c
s1

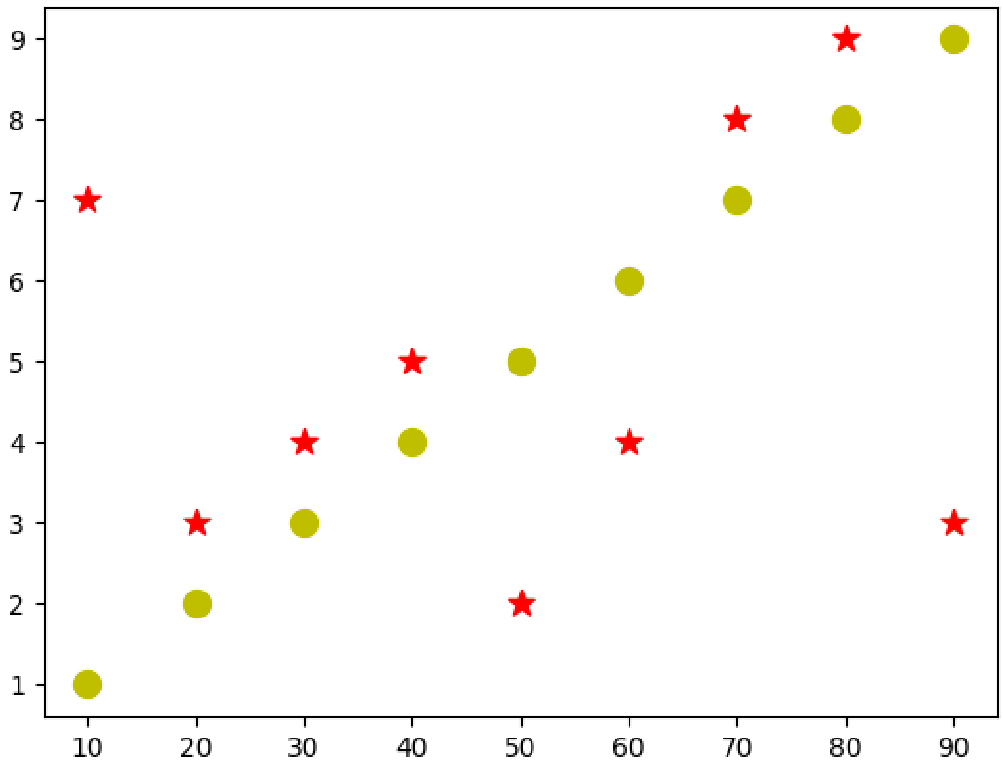


```
[ ]: x=[10,20,30,40,50,60,70,80,90]
     a=[7,3,4,5,2,4,8,9,3]
     plt.scatter(x,a)
     plt.show()
```

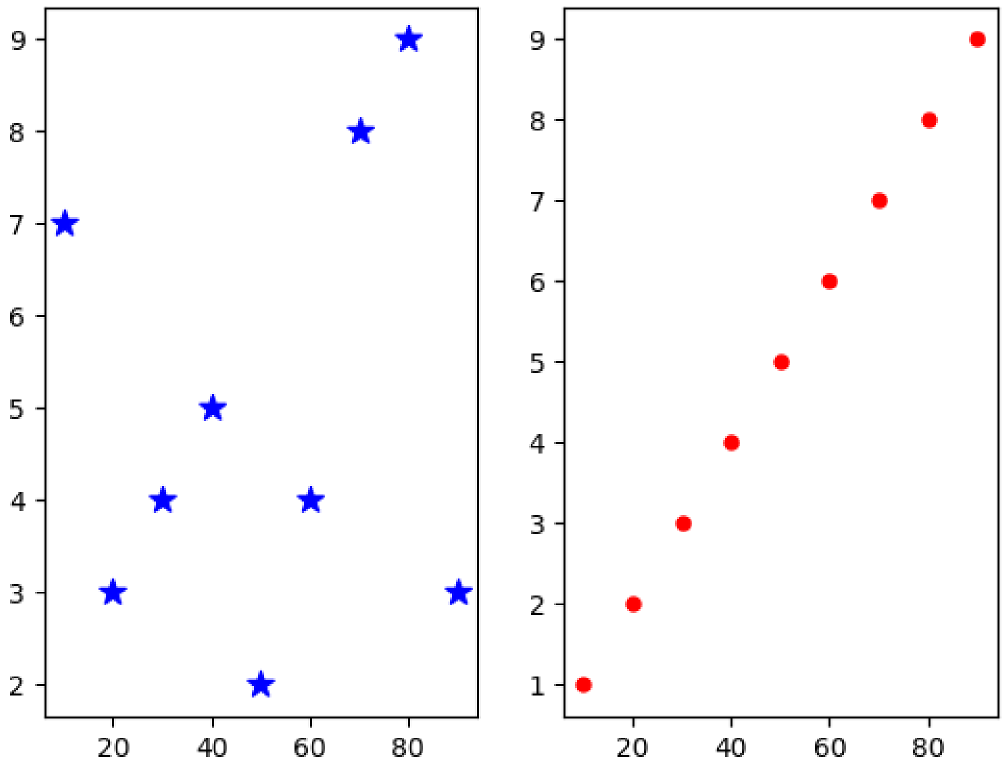


[ ]:

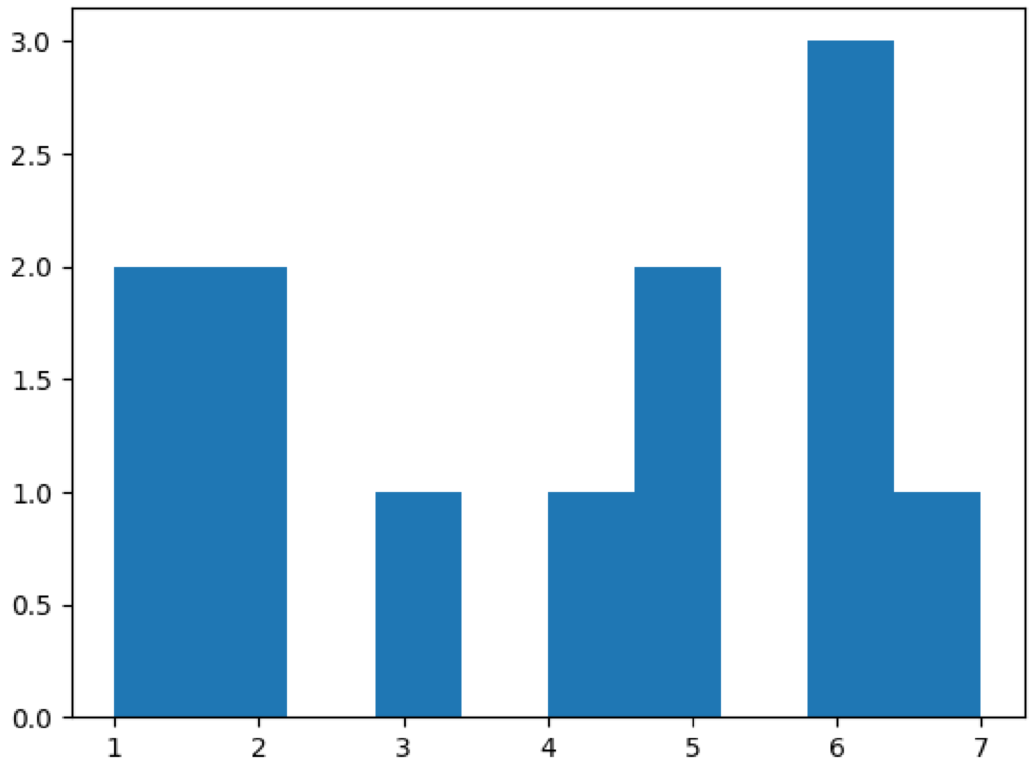
```
b=[1,2,3,4,5,6,7,8,9]
plt.scatter(x,a,marker="*",c='r',s=100)
plt.scatter(x,b,marker='o',c='y',s=100)
plt.show()
```



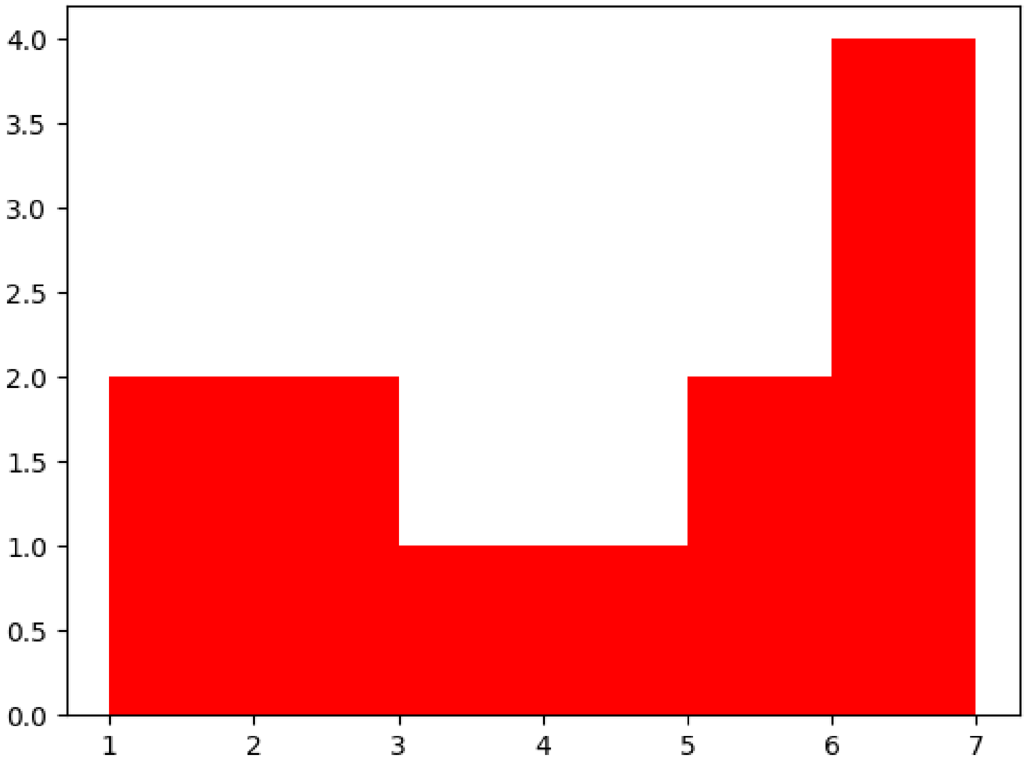
[ ]:
```
plt.subplot(1,2,1)
plt.scatter(x,a,marker="*",c="b",s=100)
plt.subplot(1,2,2)
plt.scatter(x,b,marker=".",c="r",s=100)
plt.show()
```



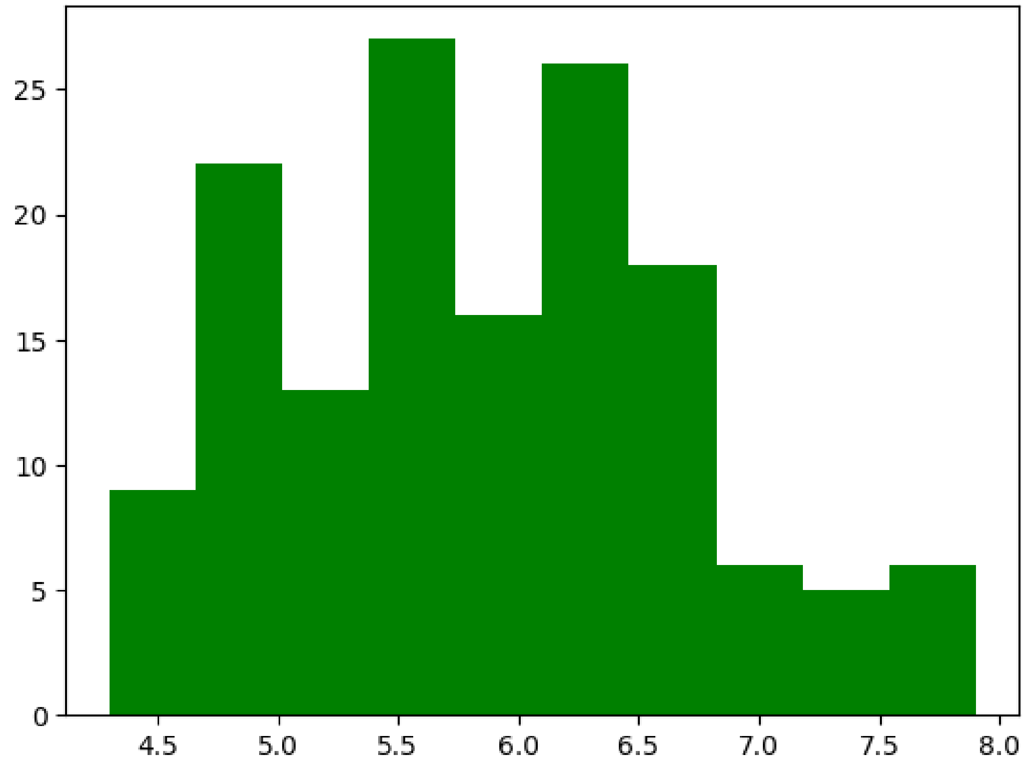
```
[ ]: data=[1,2,3,4,5,6,6,2,5,6,7,1]
     plt.hist(data)
     plt.show()
```



[ ]:
```
plt.hist(data,color="r",bins=6)
plt.show()
```

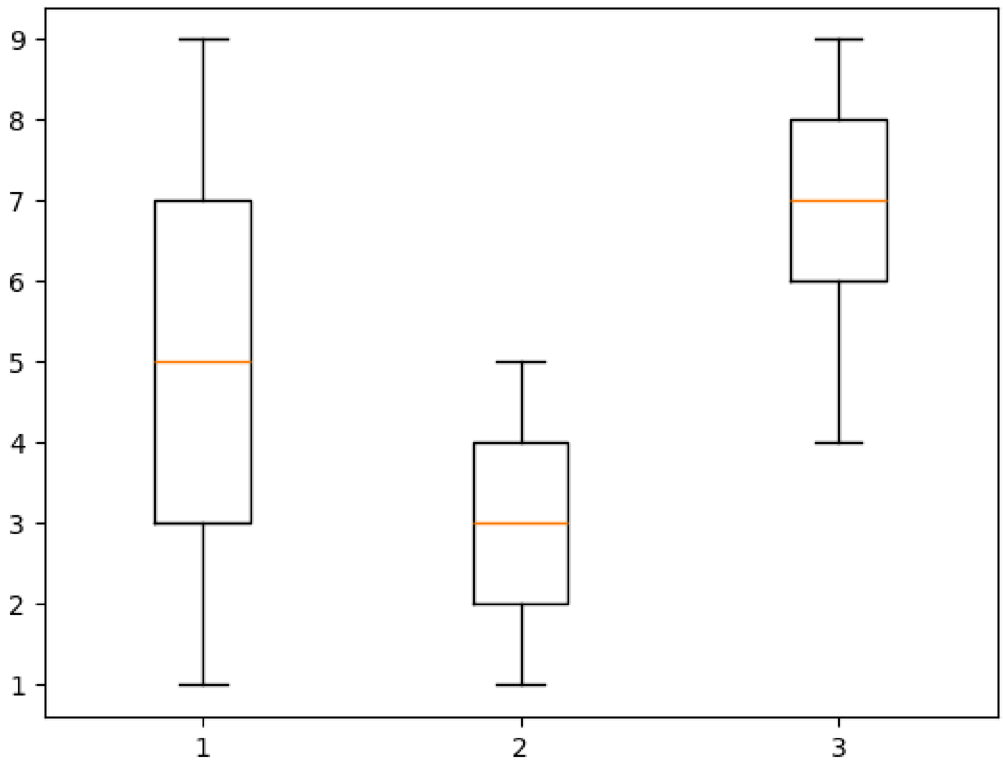


```
[ ]: #histogram of iris data - sepallength
     plt.hist(df['sepal_length'],color='g',bins=10)
     plt.show()
```

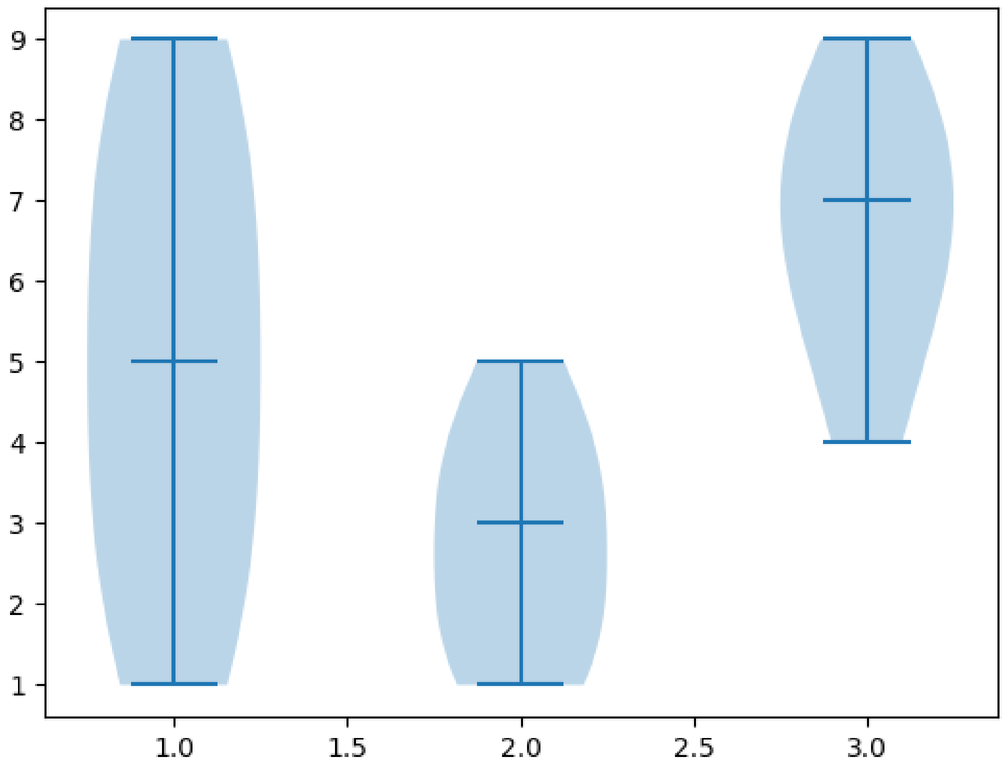


[ ]:

```
one=[1,2,3,4,5,6,7,8,9]
two=[1,2,3,4,5,4,3,2,1]
three=[6,7,8,9,8,7,6,5,4]
data=list([one,two,three])
plt.boxplot(data)
plt.show()
```

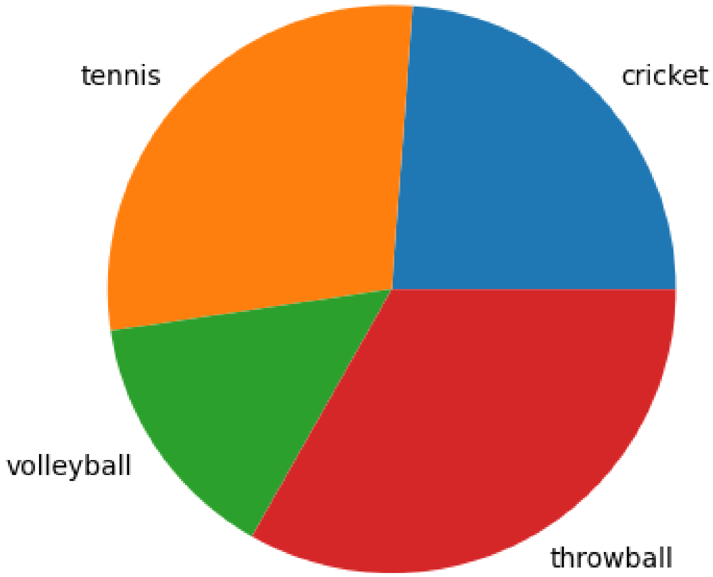


[ ]:
```
plt.violinplot(data,showmedians=True)
plt.show()
```

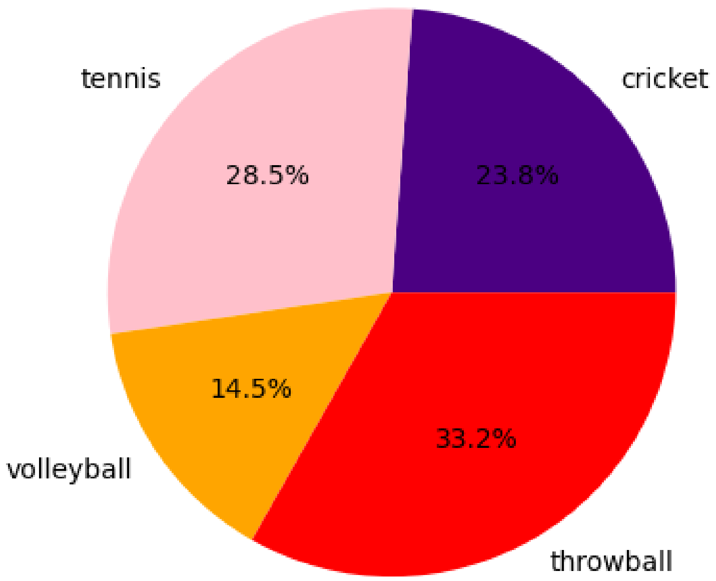


[ ]:

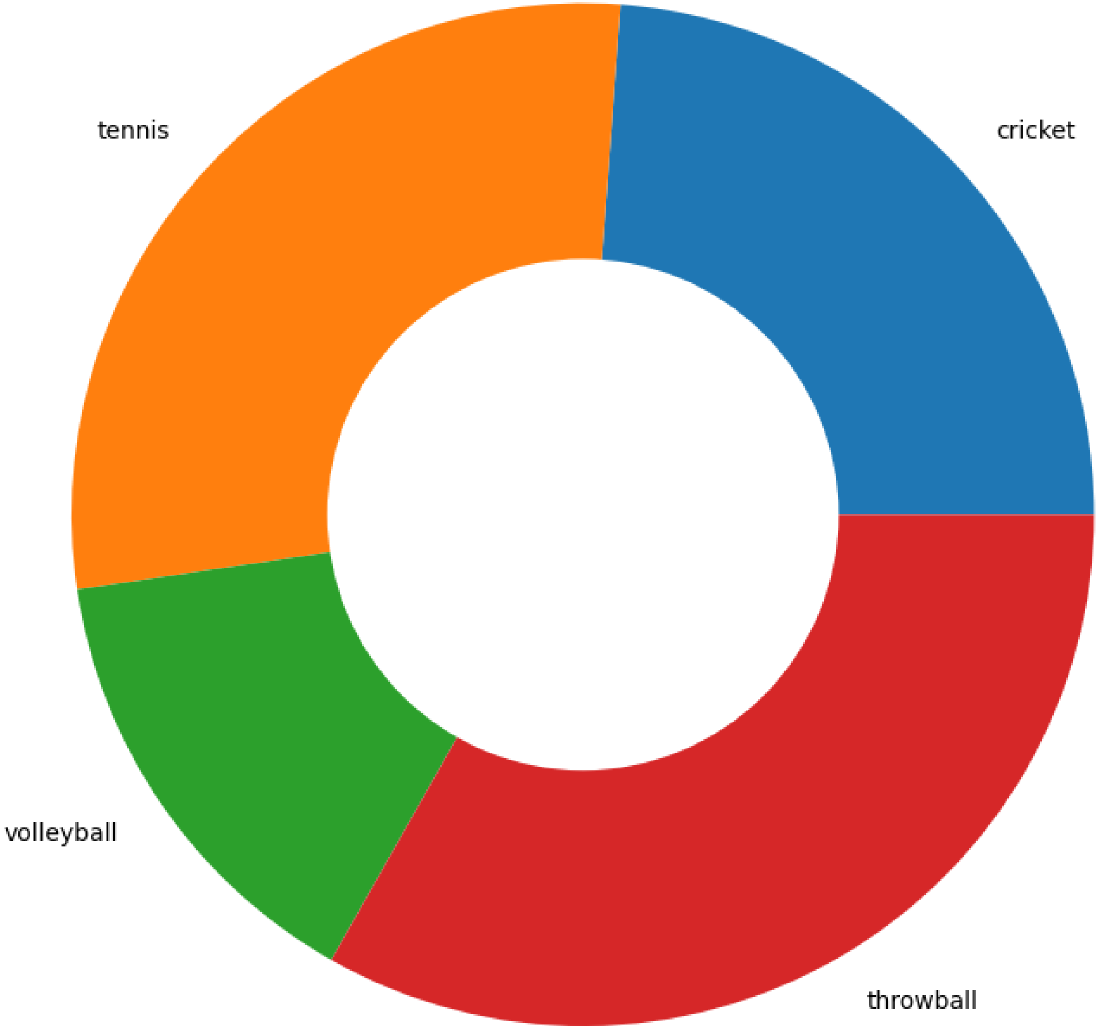
```
games=['cricket','tennis','volleyball','throwball']
quantity=[56,67,34,78]
plt.pie(quantity,labels=games)
plt.show()
```



```
[ ]: #userdefined colours piechart
     plt.pie(quantity,labels=games,autopct='%0.
       ↪1f%%',colors=['indigo','pink','orange','red'])
     plt.show()
```



```
[ ]: #doughnut chart
     plt.pie(quantity,labels=games,radius=2)
     plt.pie([1],colors=['w'],radius=1)
     plt.show()
```



5 SEABORN

```
[ ]: import seaborn as sns
```

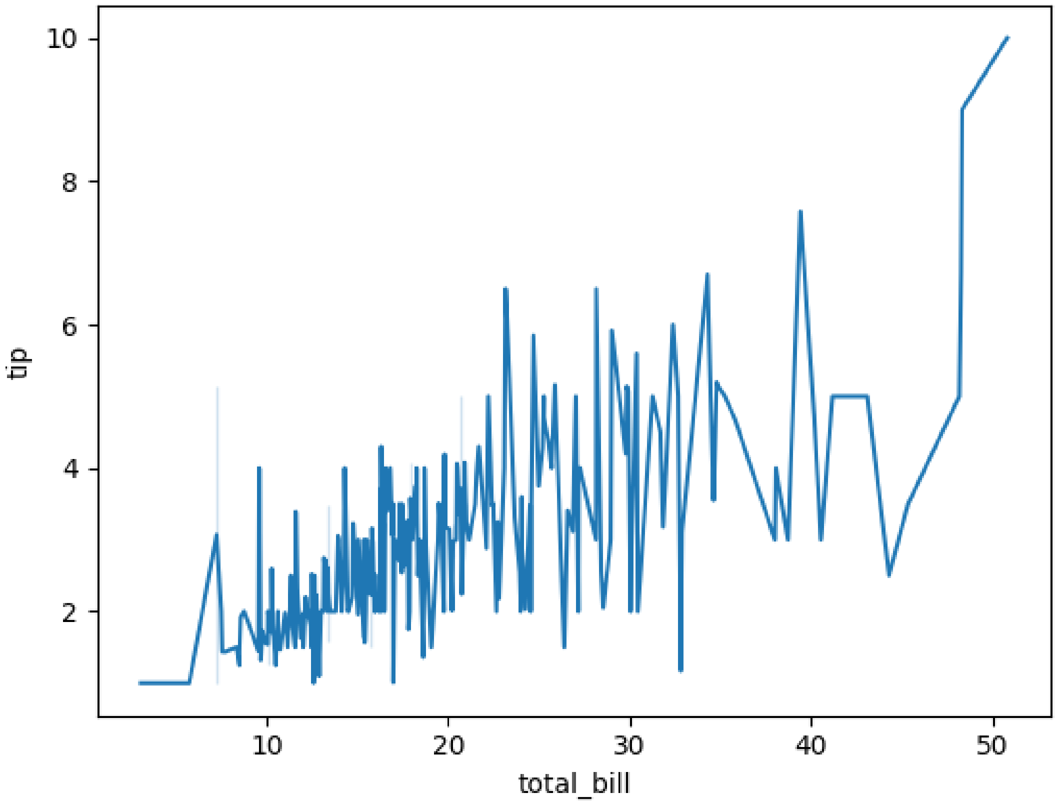
```
[ ]: print(sns.get_dataset_names())
```

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

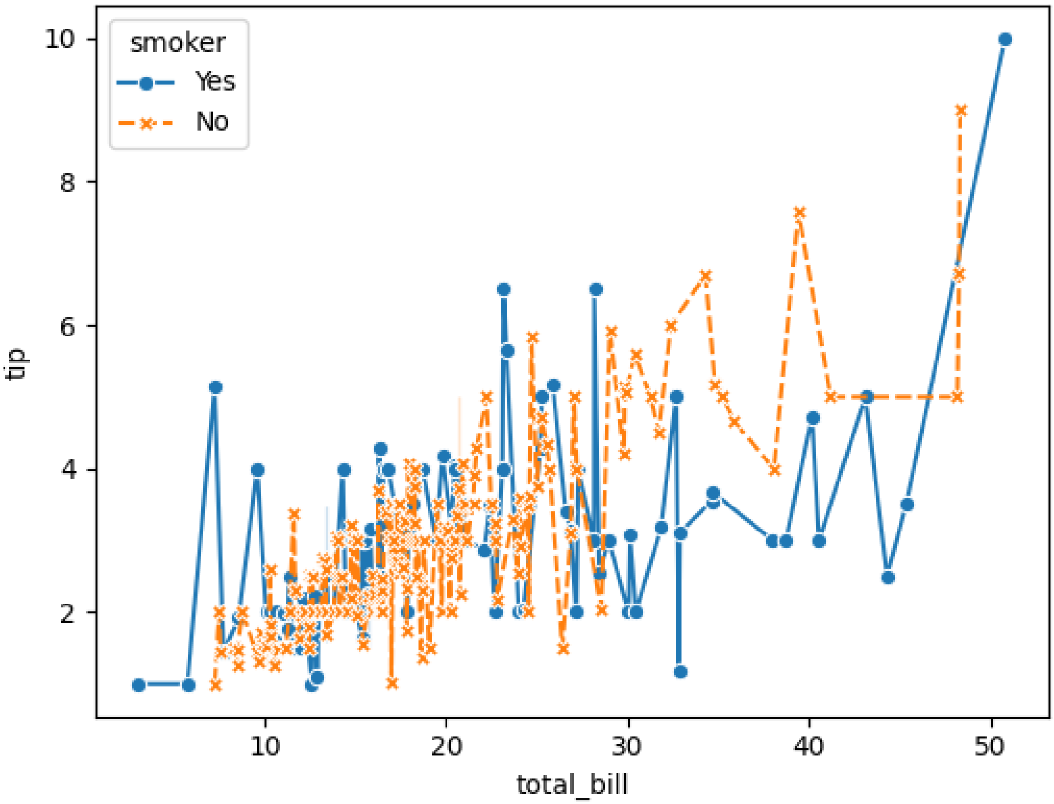
```
[ ]: sb=sns.load_dataset('tips')
     sb.head()
```

```
[ ]:    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
```

```
[ ]: sns.lineplot(x='total_bill',y='tip',data=sb)
     plt.show()
```



```
[ ]: sns.
      ↪lineplot(x='total_bill',y='tip',hue='smoker',style='smoker',markers=True,data=sb)
     plt.show()
```

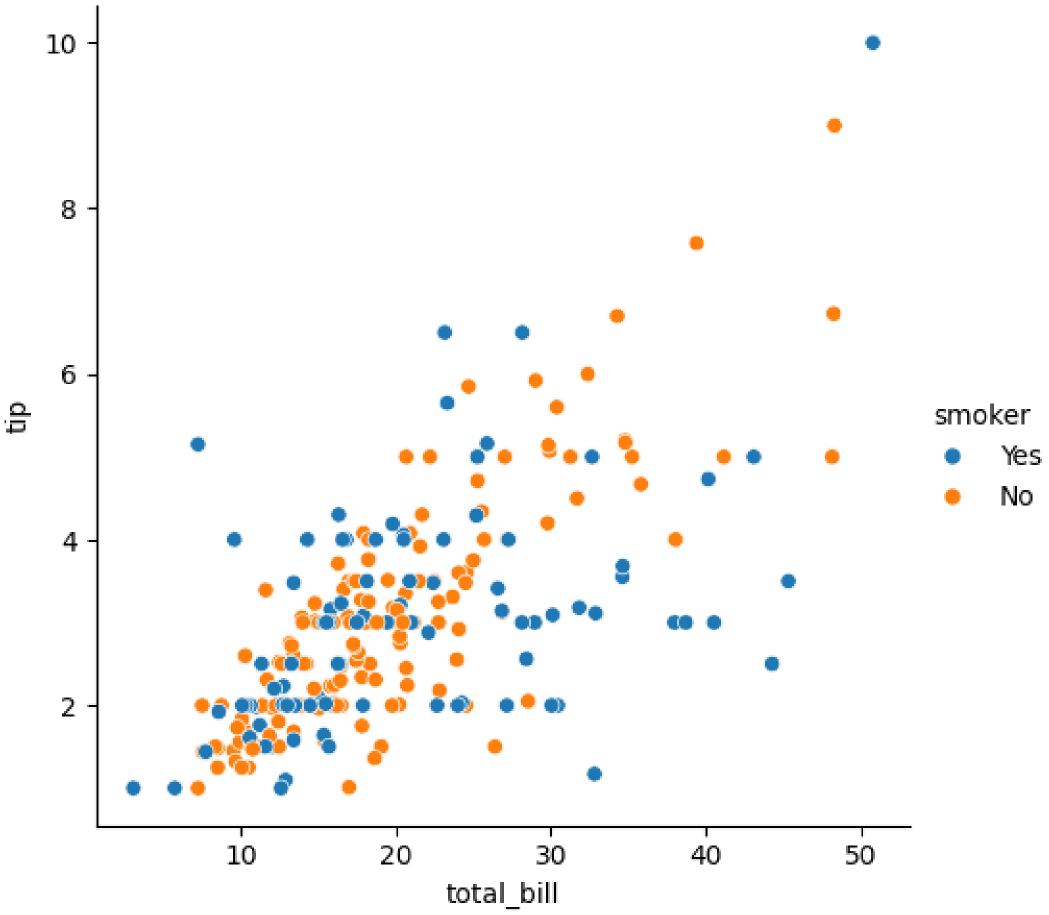


[ ]:
```
sb.shape
```

[ ]:
```
(244, 7)
```

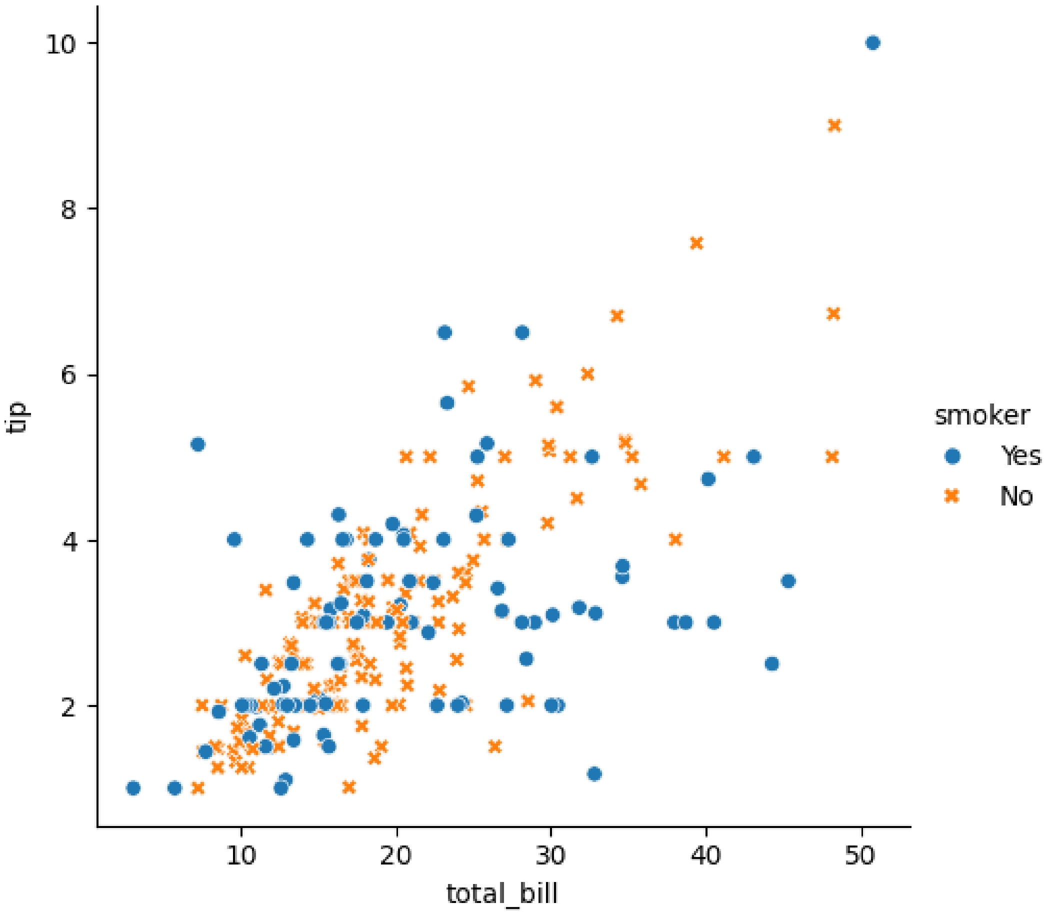
[ ]:
```
#visualizing statistical relationships
sns.relplot(data=sb,x='total_bill',y='tip',hue='smoker')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7fb57a0bda30>
```



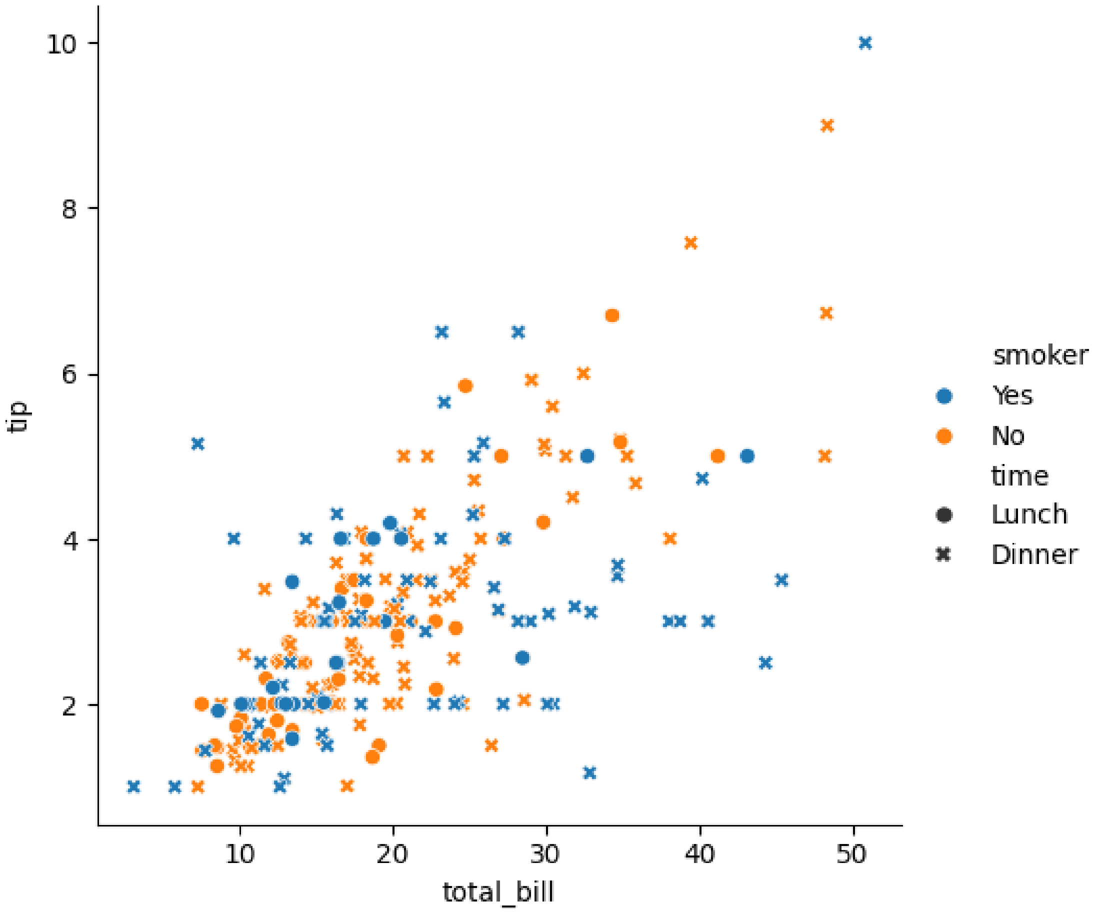
```
[ ]: #marker
     sns.relplot(data=sb,x='total_bill',y='tip',hue='smoker',style='smoker')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb57981dbb0>
```



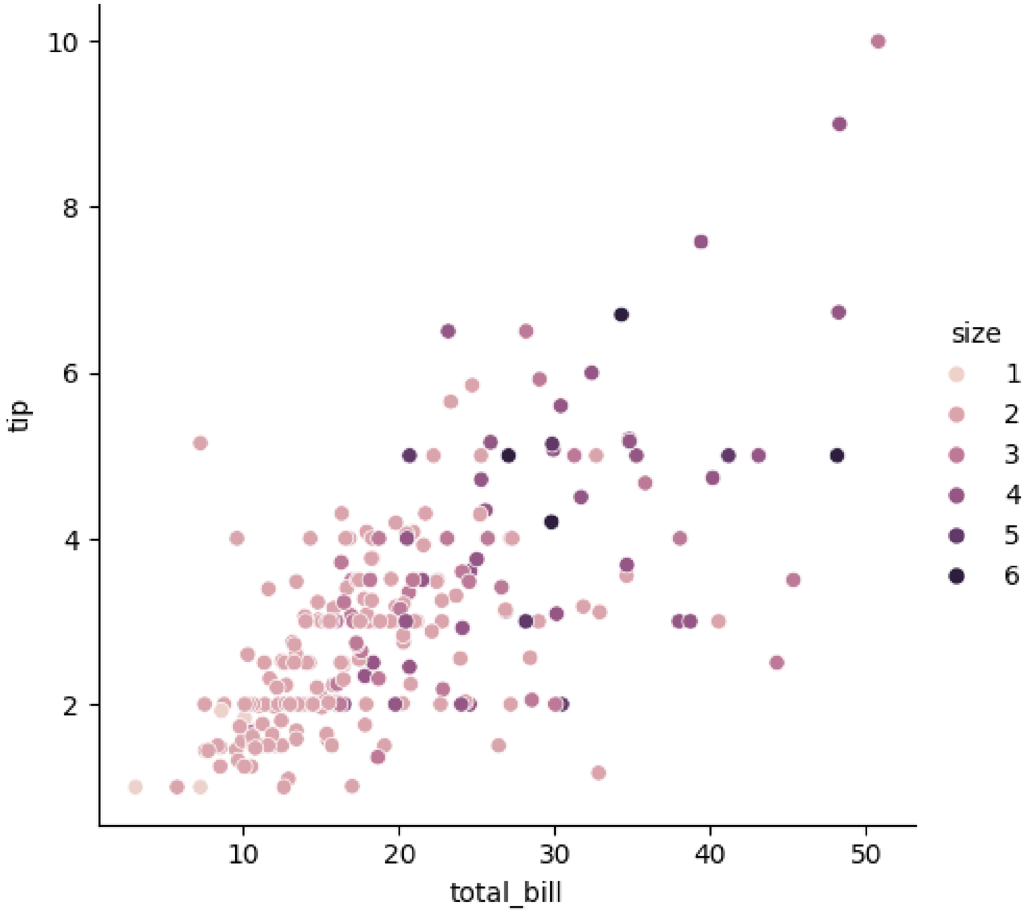
```
[ ]: sns.relplot(data=sb,x="total_bill",y="tip",hue="smoker",style="time")
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb578cc2870>
```



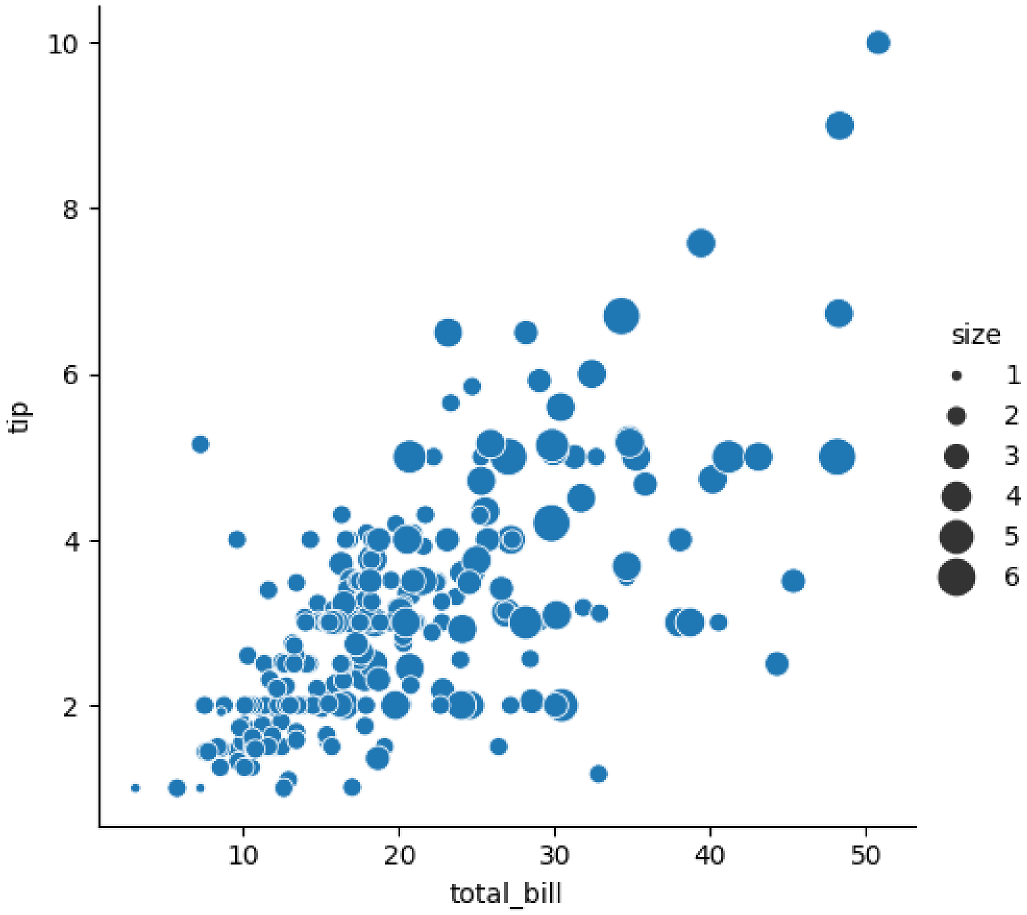
```
[ ]: sns.relplot(data=sb,x="total_bill",y="tip",hue="size",)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb57981cd40>
```



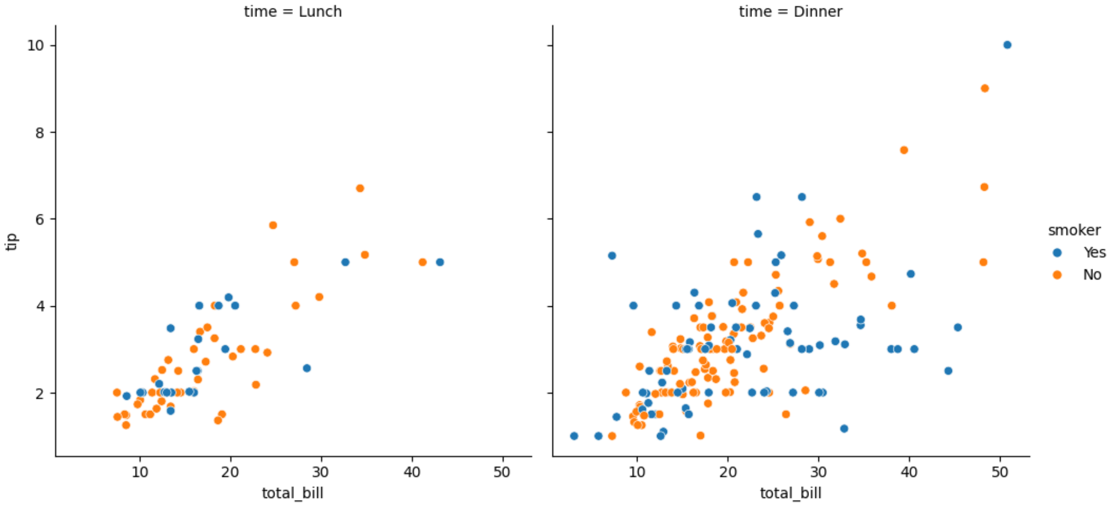
```
[ ]: sns.relplot(data=sb,x="total_bill",y="tip",size="size",sizes=(15,200))
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56f66dc70>
```



```
[ ]: sns.relplot(data=sb,x="total_bill",y="tip",hue="smoker",col="time",)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb579d01a30>
```



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

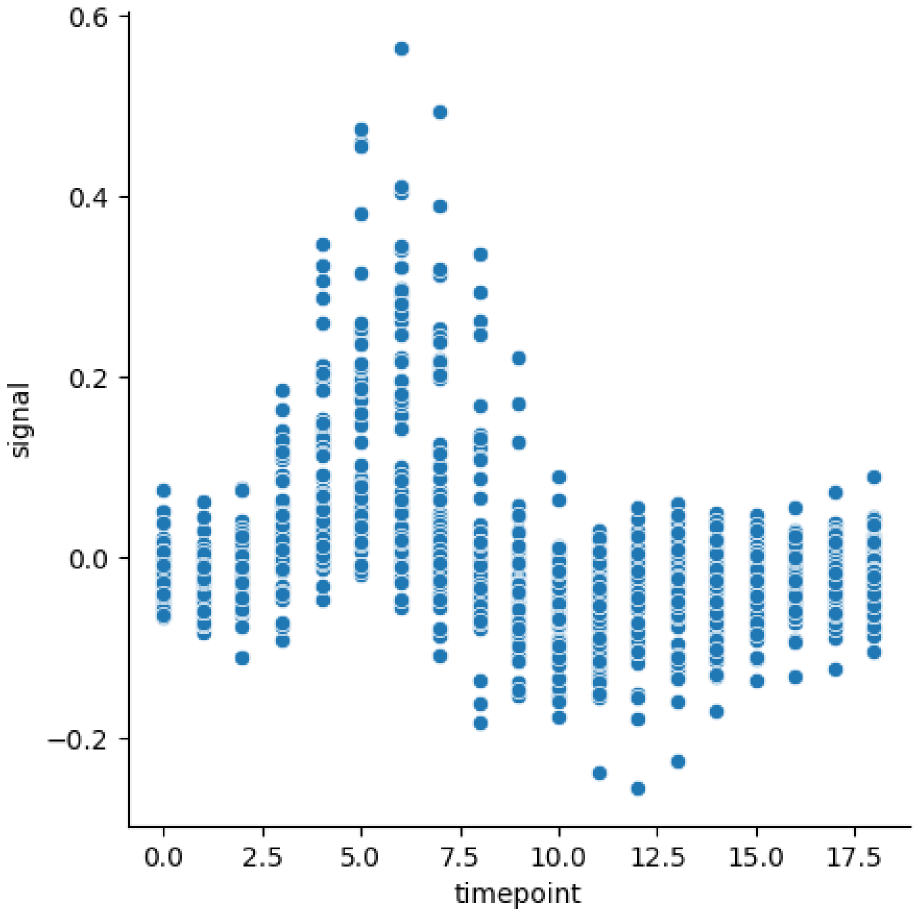
```
[ ]:   subject  timepoint event    region    signal
     0     s13         18  stim  parietal -0.017552
     1      s5         14  stim  parietal -0.080883
     2     s12         18  stim  parietal -0.081033
     3     s11         18  stim  parietal -0.046134
     4     s10         18  stim  parietal -0.037970
```

```
[ ]: fmri.shape
```

```
[ ]: (1064, 5)
```

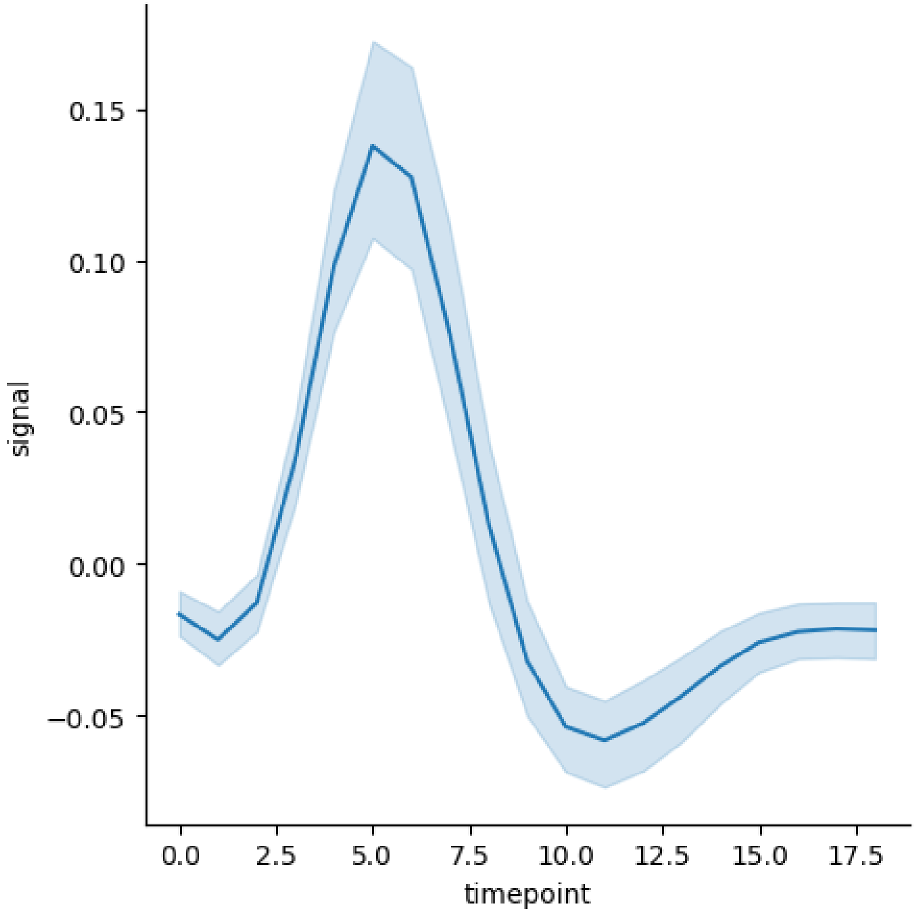
```
[ ]: sns.relplot(data=fmri,x='timepoint',y='signal')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56da34d40>
```



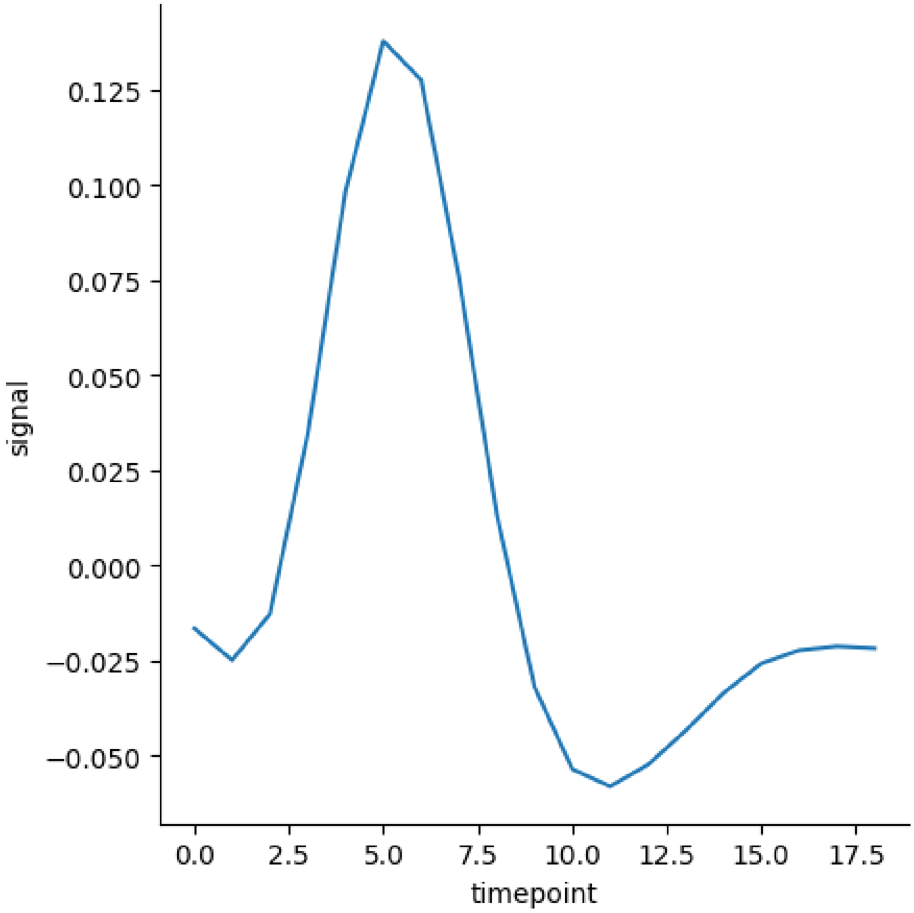
```
[ ]: sns.relplot(data=fmri,x='timepoint',y='signal',kind='line')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56db13320>
```



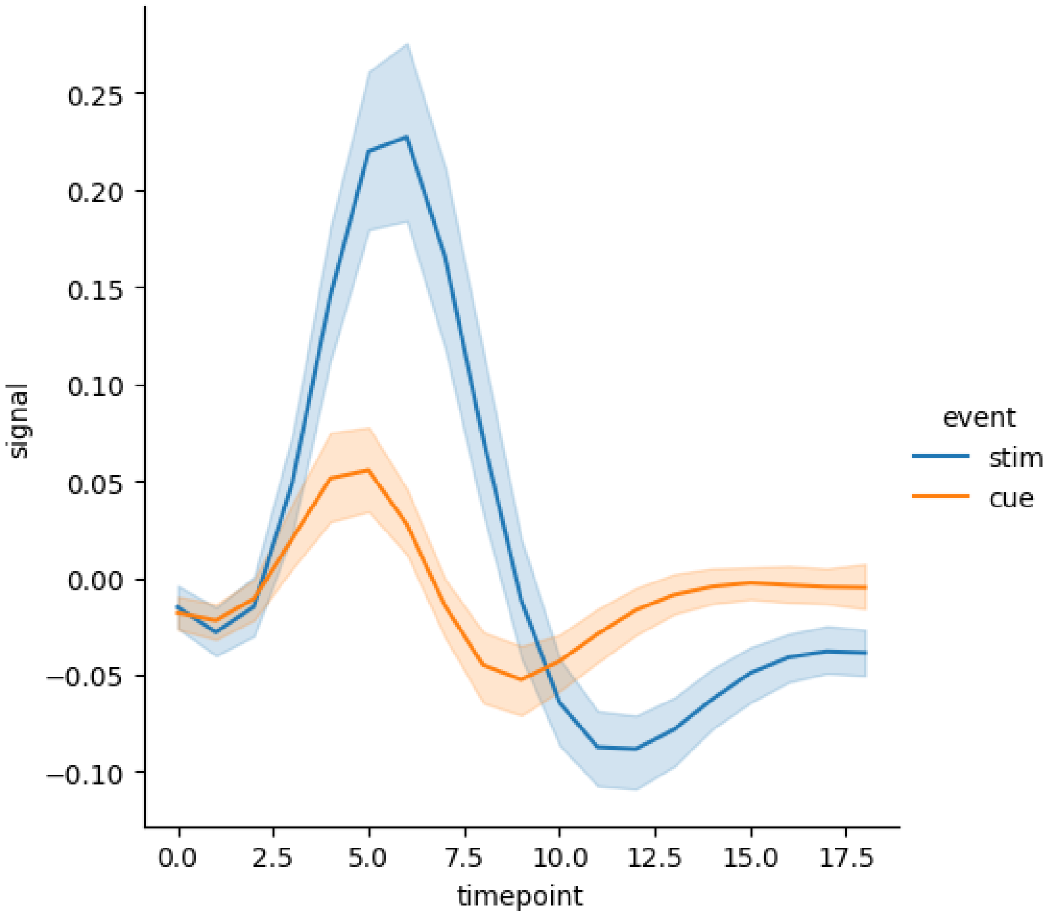
```
[ ]: #remove error band
     sns.relplot(data=fmri,kind='line',x='timepoint',y='signal',errorbar=None)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56f675c70>
```



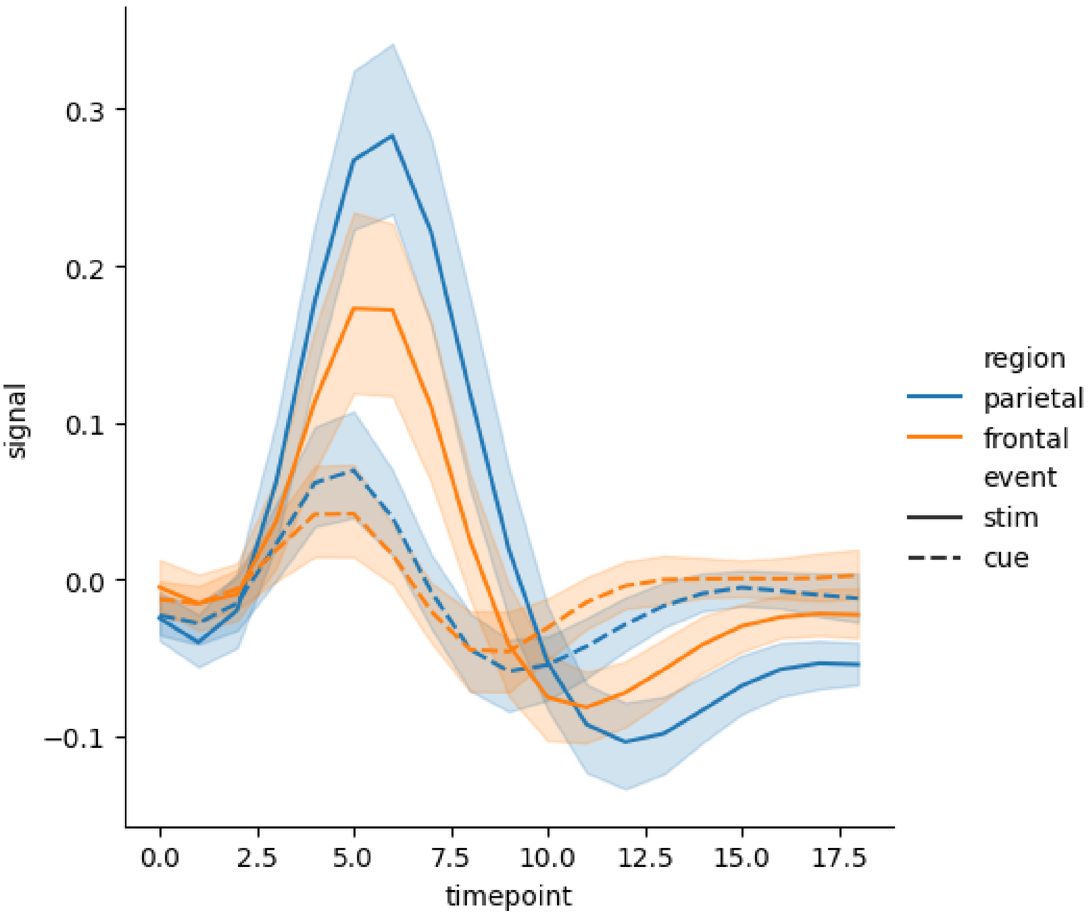
```
[ ]: #plotting into two lines and error bands
     sns.relplot(data=fmri,kind='line',x='timepoint',y='signal',hue='event')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56f674d40>
```



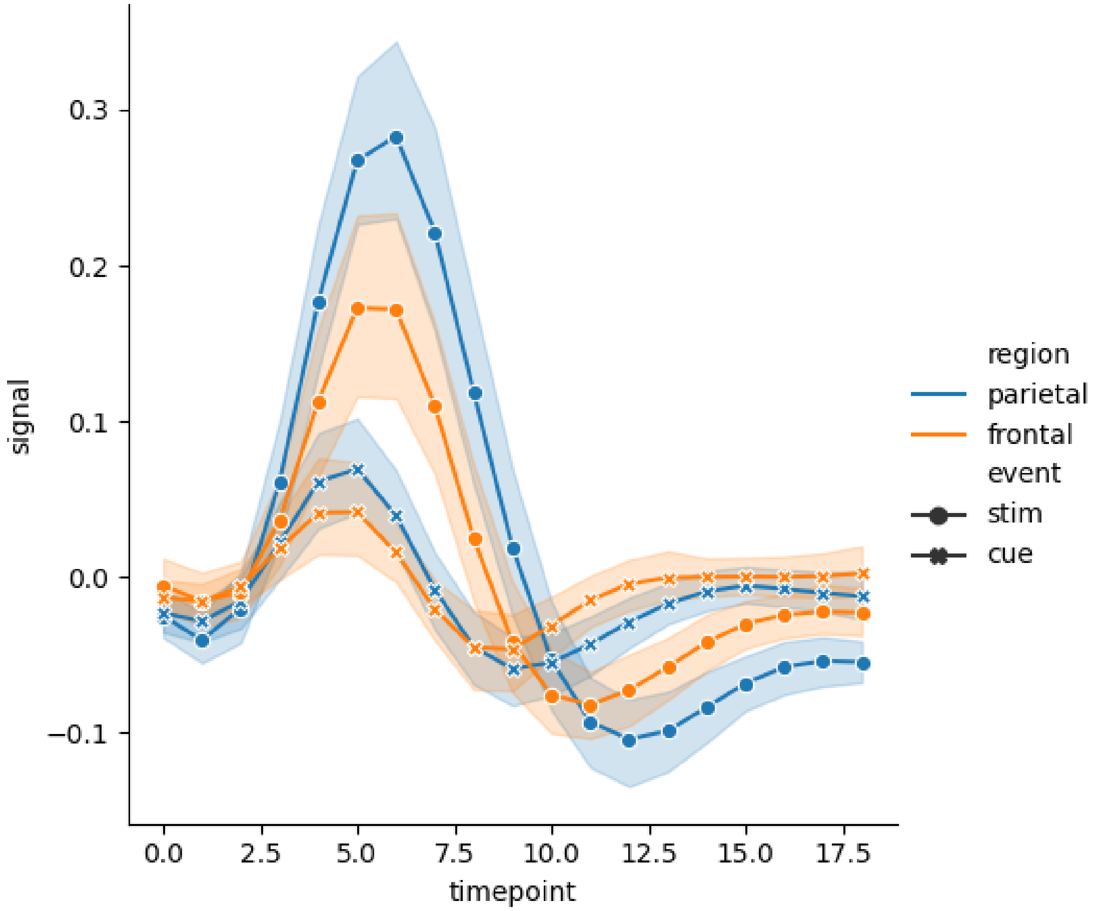
```
[ ]: sns.
      ↪relplot(data=fmri,kind='line',x='timepoint',y='signal',hue='region',style='event')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56da75250>
```



```
[ ]: sns.
      ↪relplot(data=fmri,kind='line',x='timepoint',y='signal',hue='region',style='event',dashes=Fa
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56dabcad0>
```

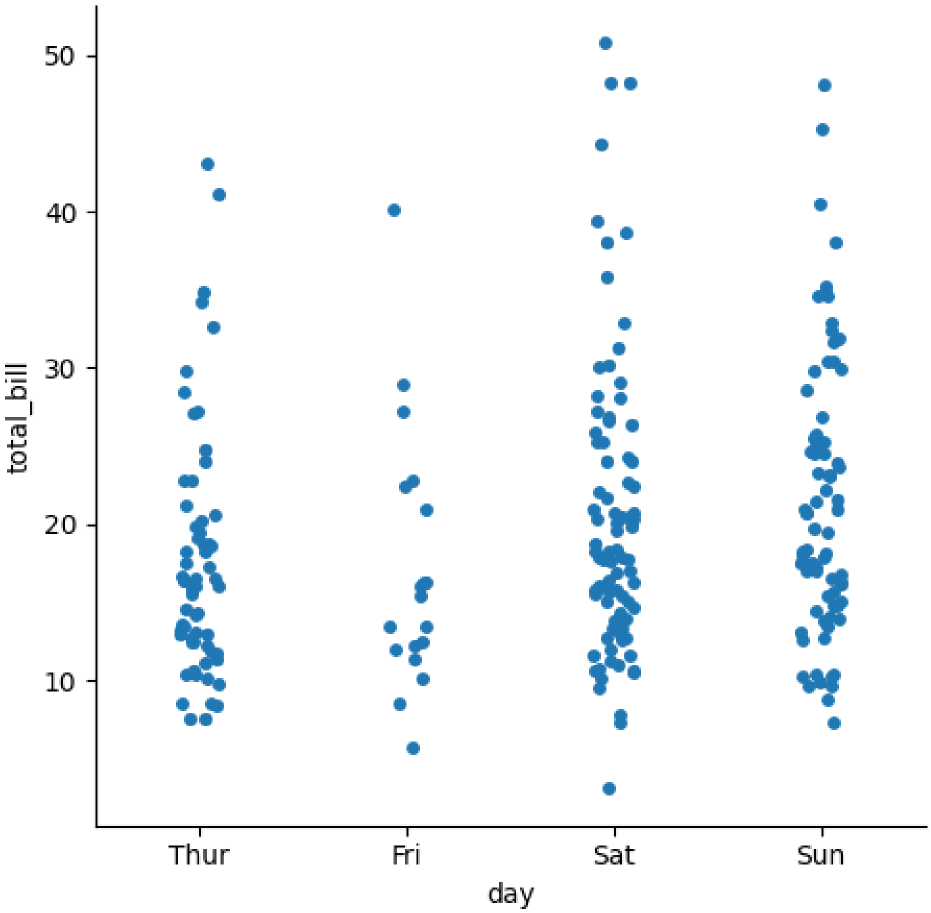


```
[ ]: tips=sns.load_dataset('tips')
     tips.head()
```

```
[ ]:    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
```

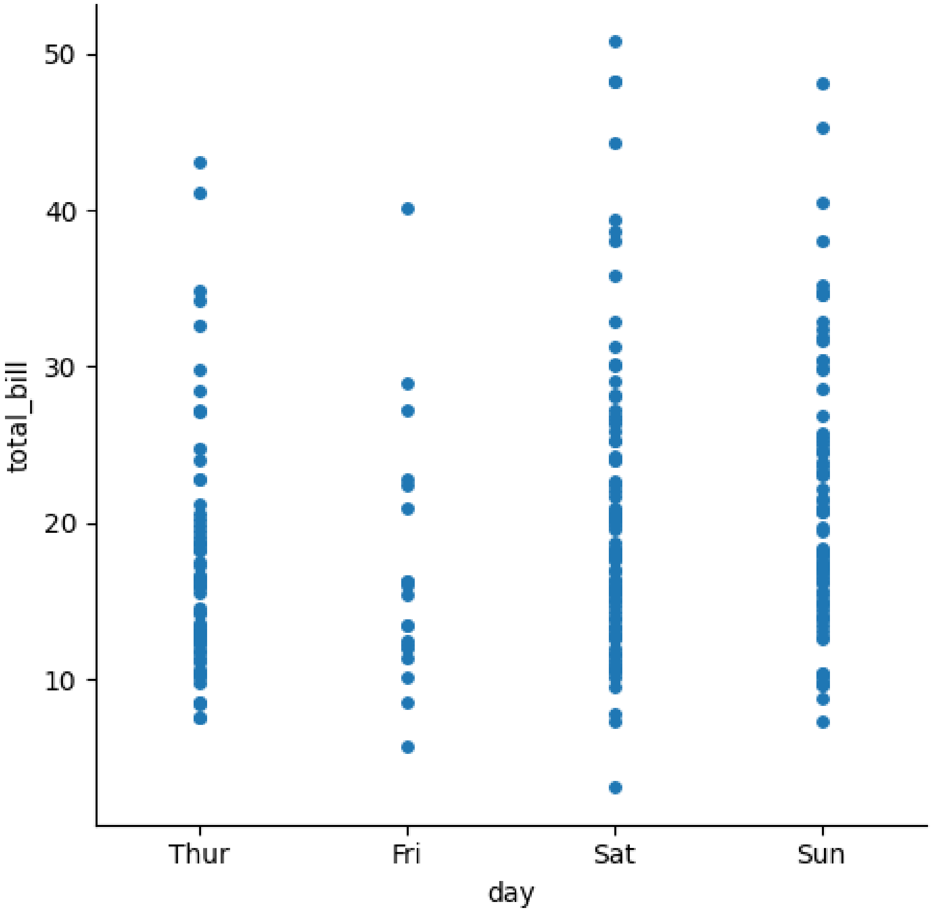
```
[ ]: #visualizing categorical data
     sns.catplot(data=tips,x='day',y='total_bill')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56f438e00>
```



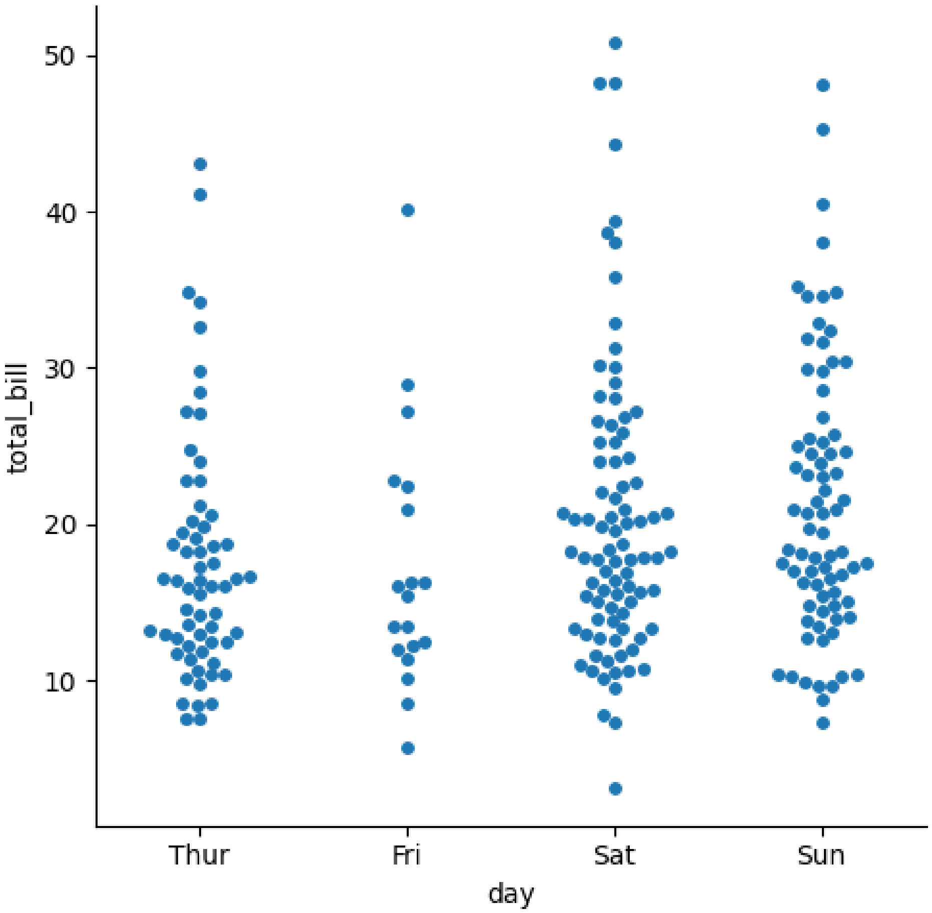
```
[ ]: #jitter
     sns.catplot(data=tips,x='day',y='total_bill',jitter=False)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56d798470>
```



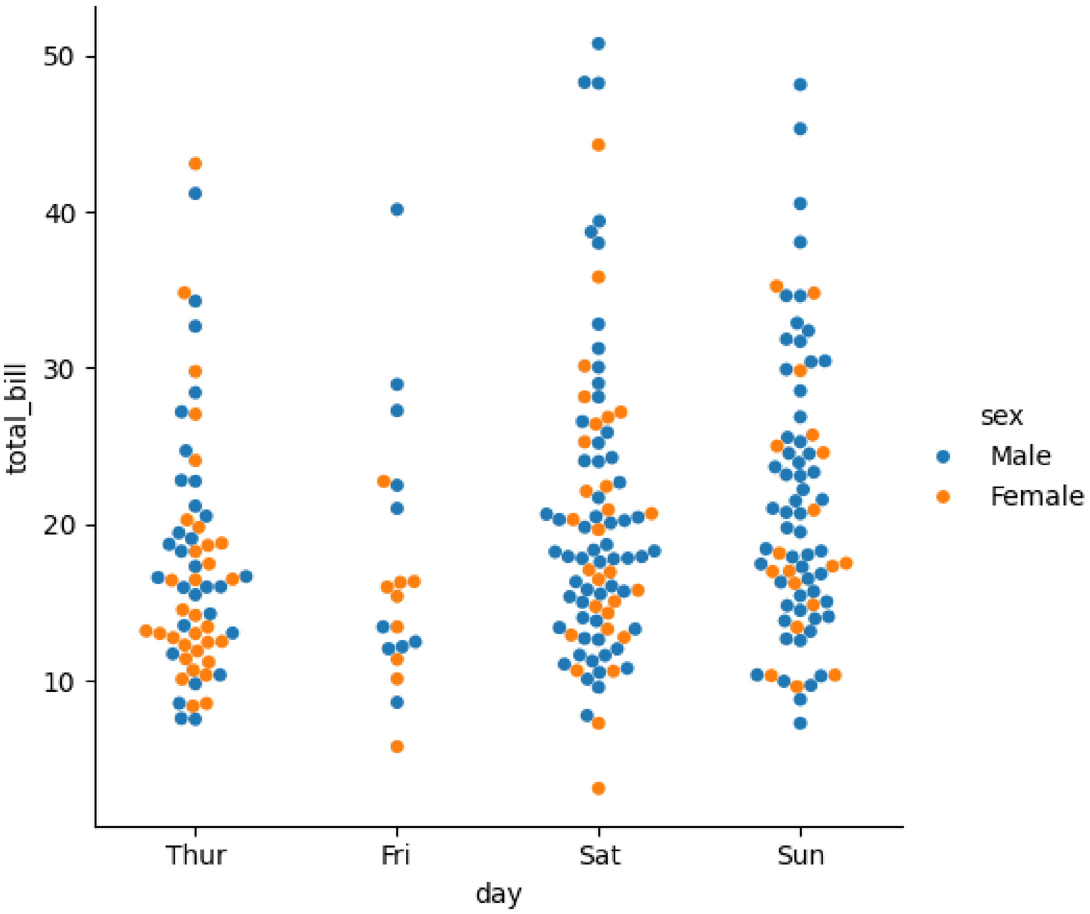
```
[ ]: #swarm plot
     sns.catplot(data=tips,x='day',y='total_bill',kind='swarm')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb57995bd70>
```



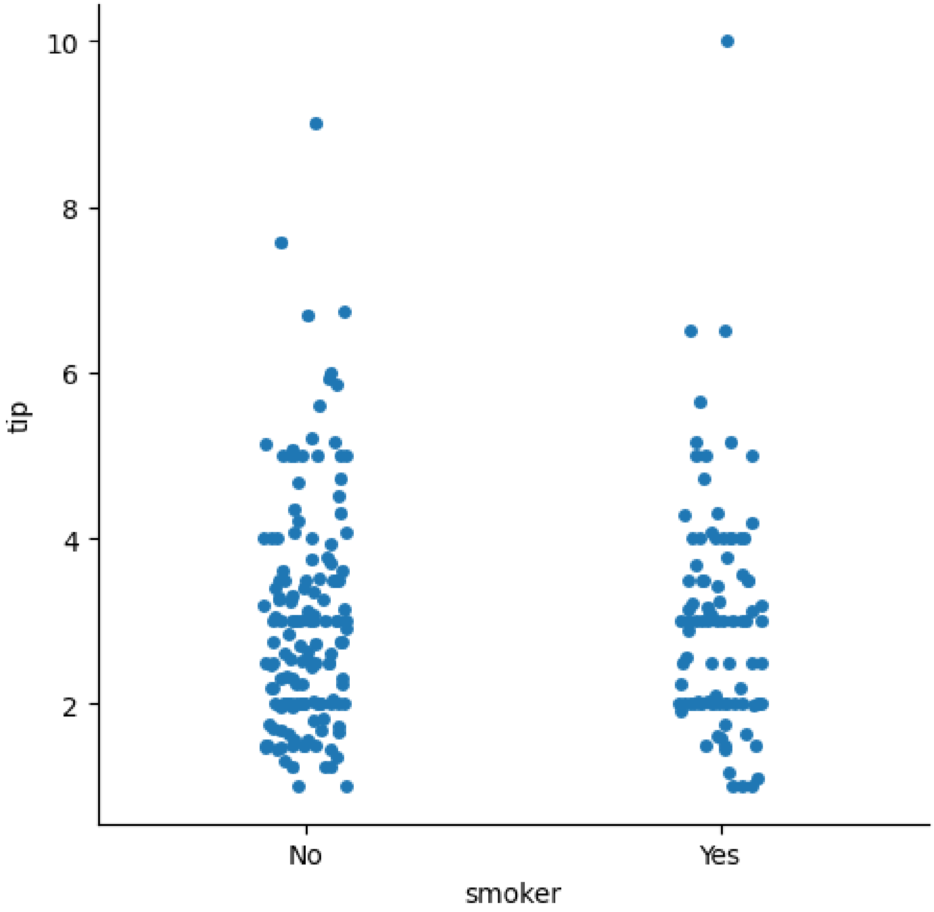
[ ]:
```
sns.catplot(data=tips,x='day',y='total_bill',hue='sex',kind='swarm')
```

[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x7fb5793da210>
```



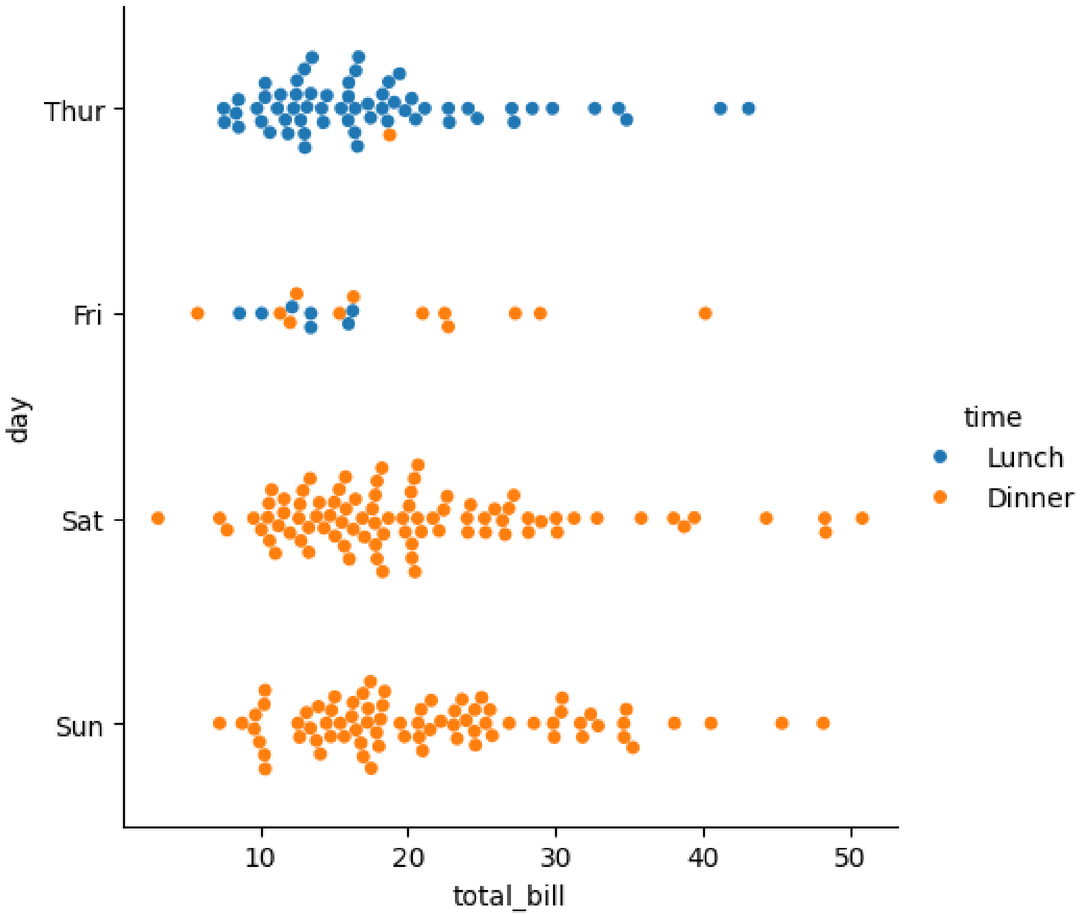
```
[ ]: sns.catplot(data=tips,x='smoker',y='tip',order=['No','Yes'])
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb584425e80>
```



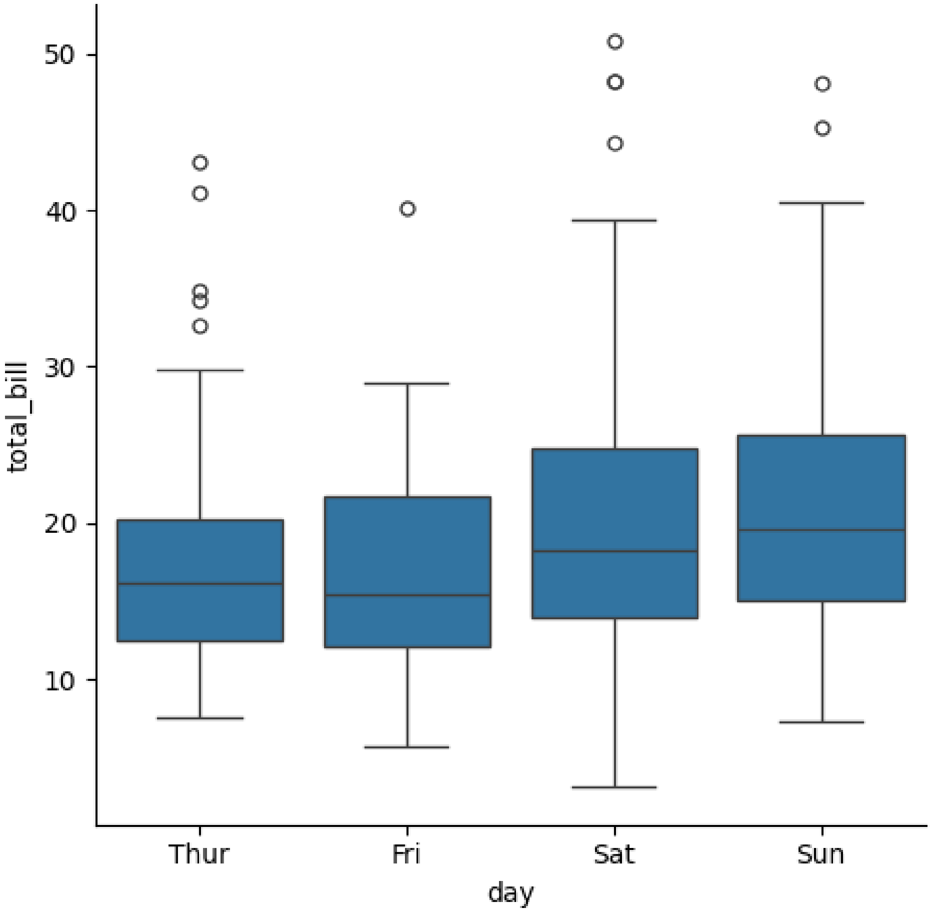
```
[ ]: sns.catplot(data=tips,x='total_bill',y='day',hue='time',kind='swarm')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56d74ecc0>
```



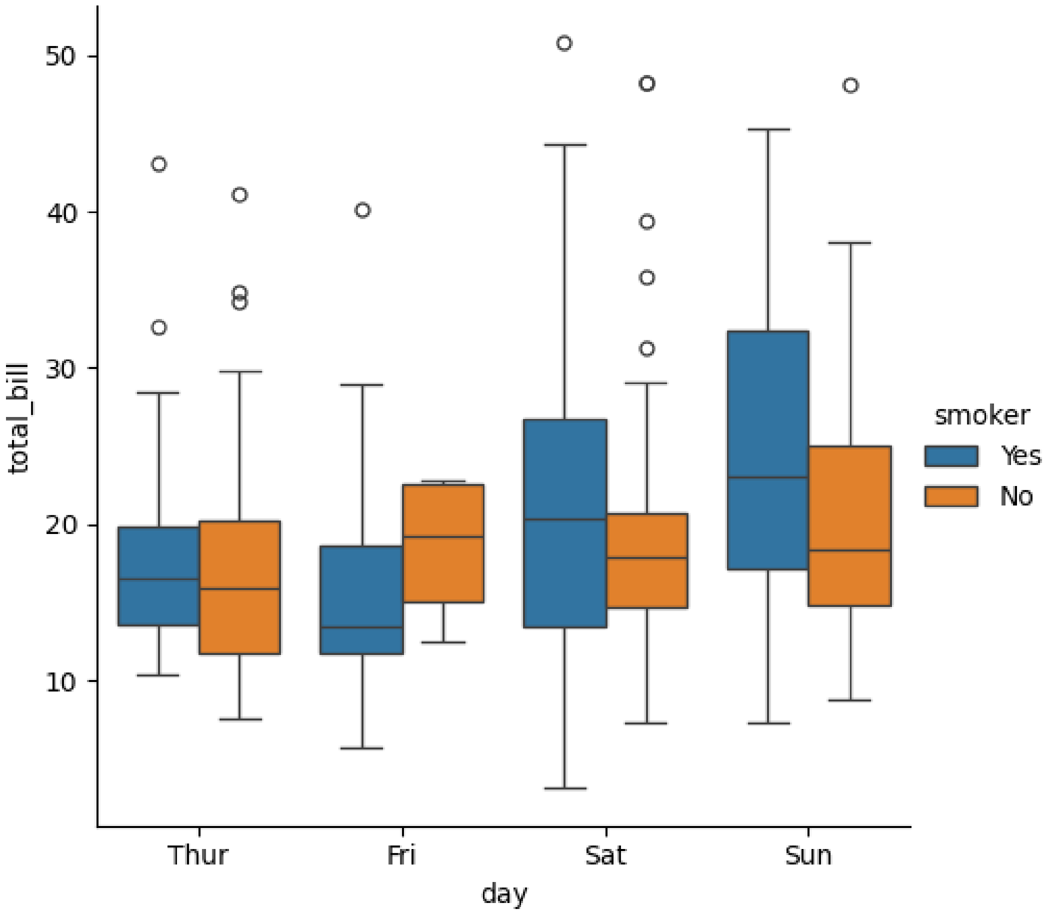
```
[ ]: sns.catplot(data=tips,x='day',y='total_bill',kind='box')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56f5070e0>
```



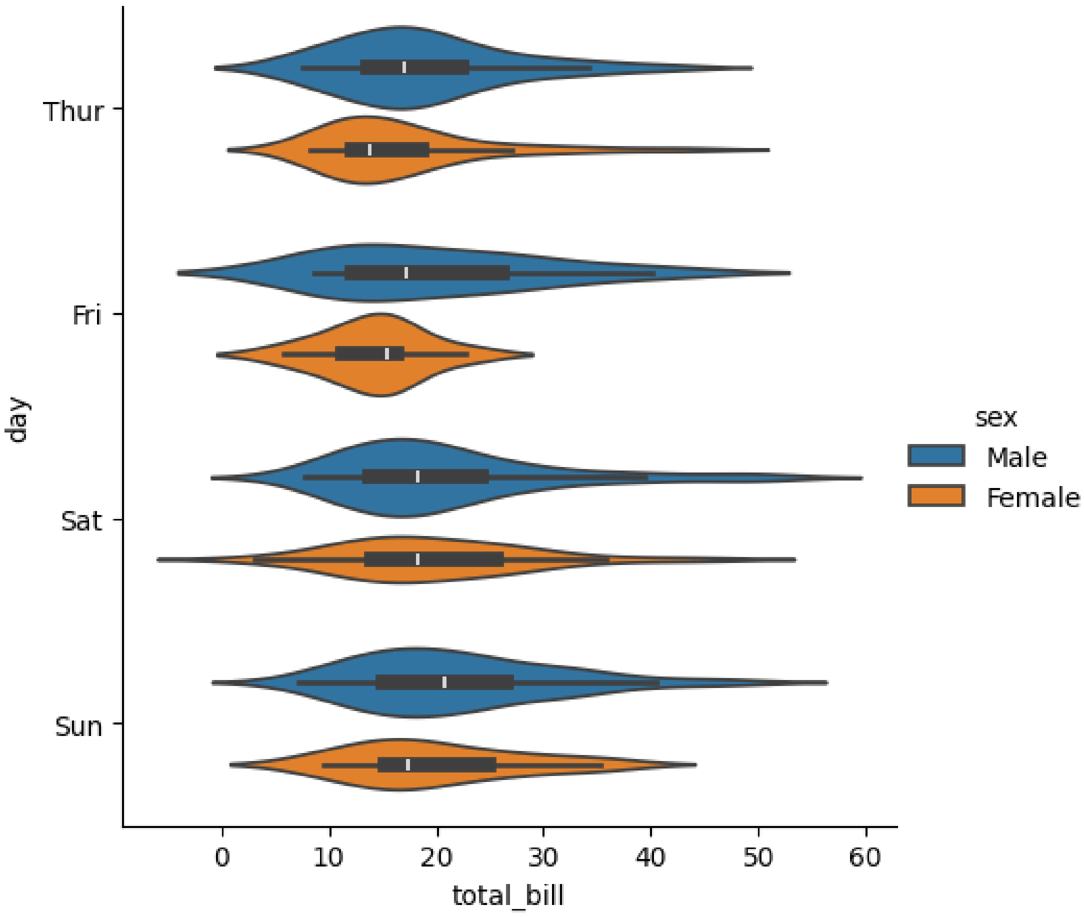
```
[ ]: sns.catplot(data=tips,x='day',y='total_bill',hue='smoker',kind='box')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56d626f00>
```



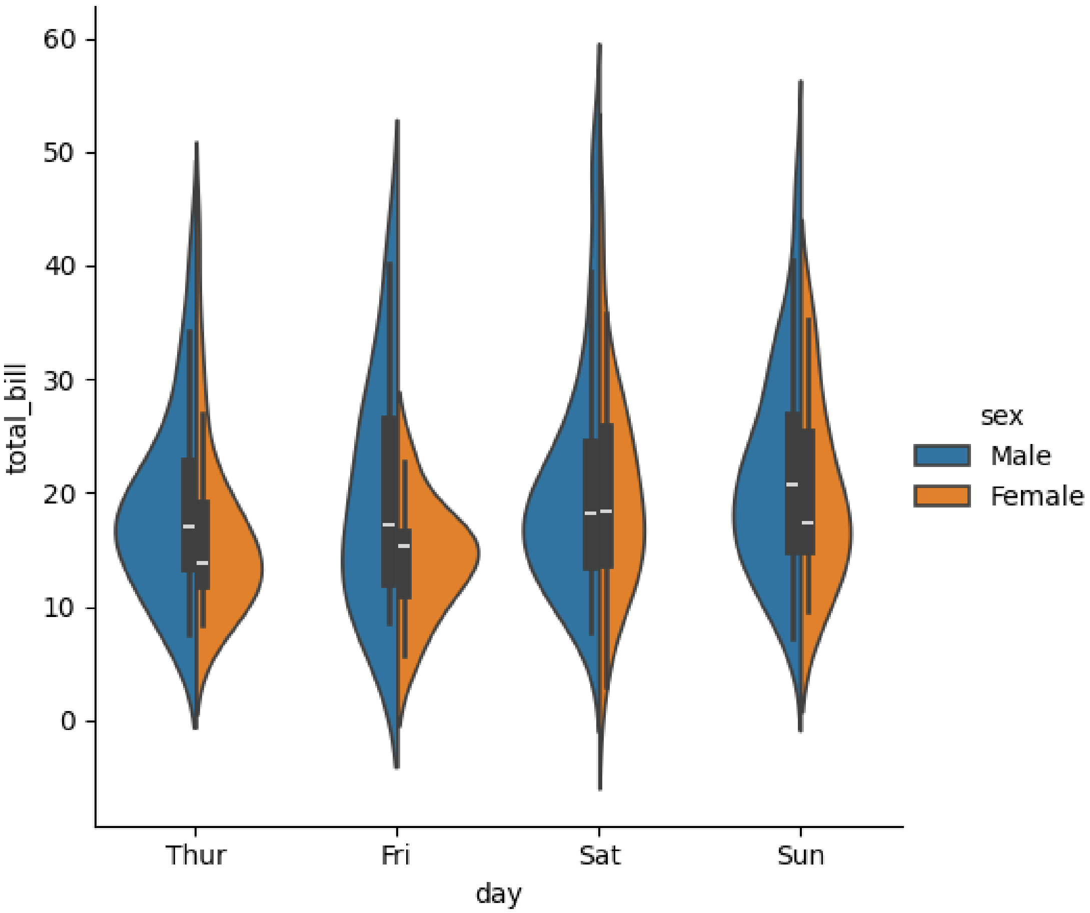
```
[ ]: sns.catplot(data=tips,x='total_bill',y='day',hue='sex',kind='violin')
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56d74f650>
```



```
[ ]: sns.catplot(data=tips,x='day',y='total_bill',hue='sex',kind='violin',split=True)
```

```
[ ]: <seaborn.axisgrid.FacetGrid at 0x7fb56d4d0c20>
```


60
50
40
30
20
10
0
total_bill
Thur
Fri
Sat
Sun
day
sex
Male
Female