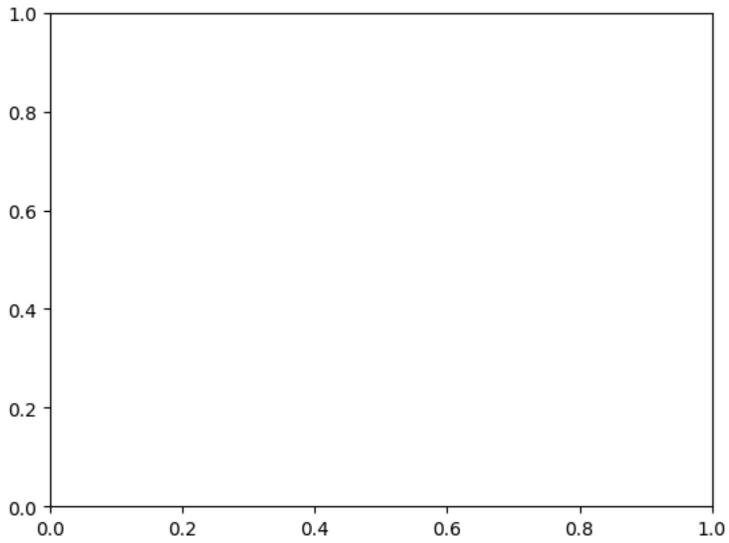
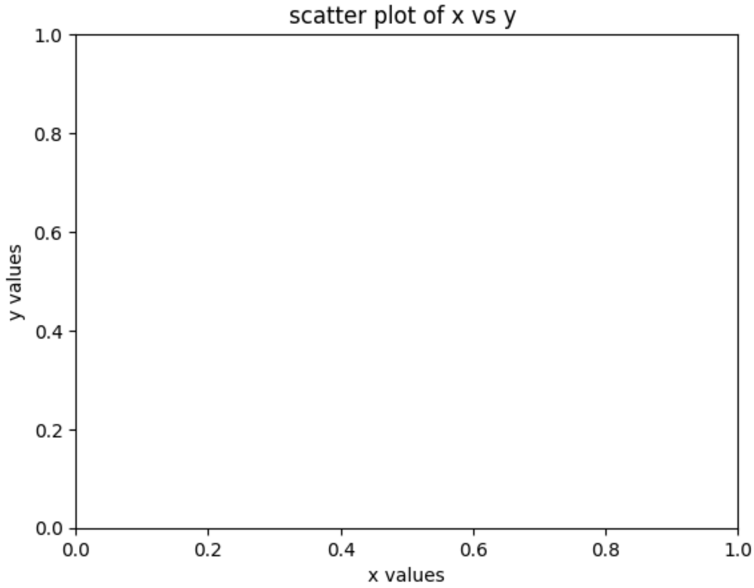
```
import matplotlib.pyplot as plt


x=[1,2,3,4,5,6,7,8,9,10]
y=[2,4,5,7,6,8,9,10,12,13]


plt.scatter(x,y,color='blue',marker='0')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/matplotlib/markers.py in _set_marker(self, marker)
    358             try:
--> 359                 Path(marker)
    360                 self._marker_function = self._set_vertices

                                    7 frames
ValueError: 'vertices' must be 2D with shape (M, 2). Your input has shape ().

The above exception was the direct cause of the following exception:

ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/matplotlib/markers.py in _set_marker(self, marker)
    360                 self._marker_function = self._set_vertices
    361             except ValueError as err:
--> 362                 raise ValueError('Unrecognized marker style {!r}'
    363                                  .format(marker)) from err
    364

ValueError: Unrecognized marker style '0'
```

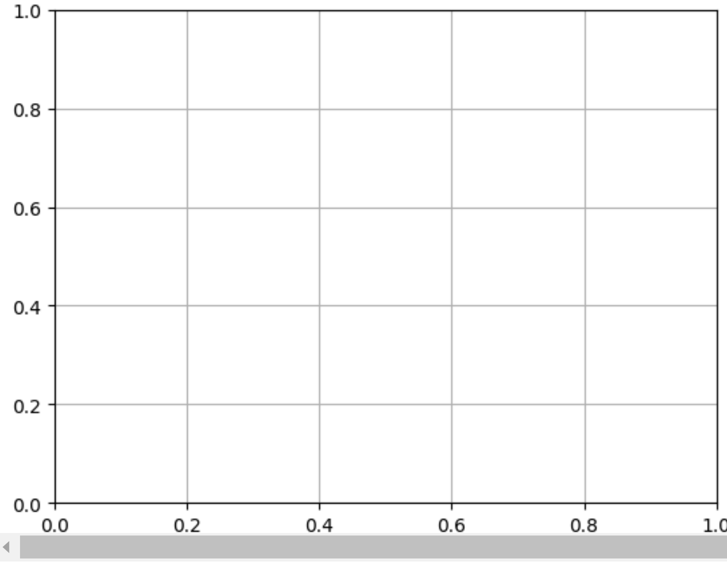


```
plt.title('scatter plot of x vs y ')
plt.xlabel('x values')
plt.ylabel('y values')
```

```
Text(0, 0.5, 'y values')
```



```
plt.grid()
```



```
plt.show()
```

```
#question no 2  data vis assignment
```

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

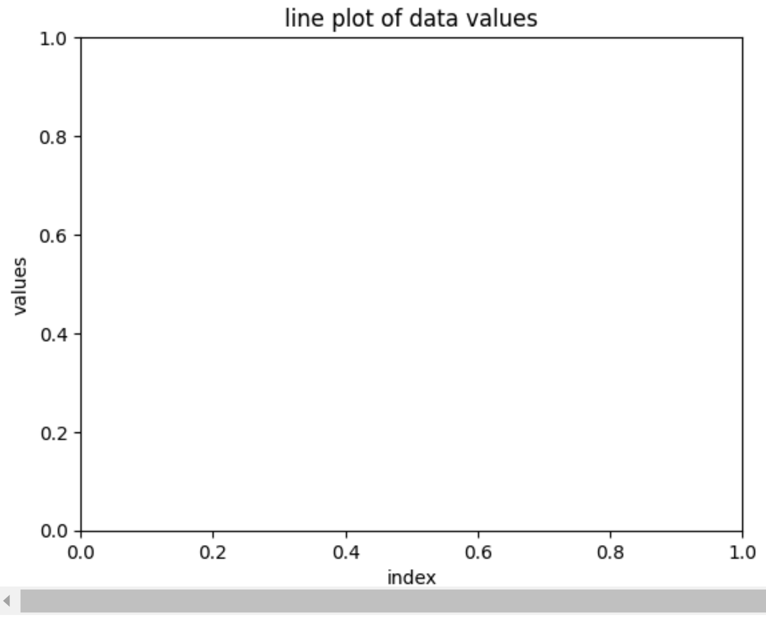
```
#given data
data=np.array([3,7,9,15,27,29,35])
```

```
plt.plot(x,data,maker='0',color='blue',linestyle='_')
```

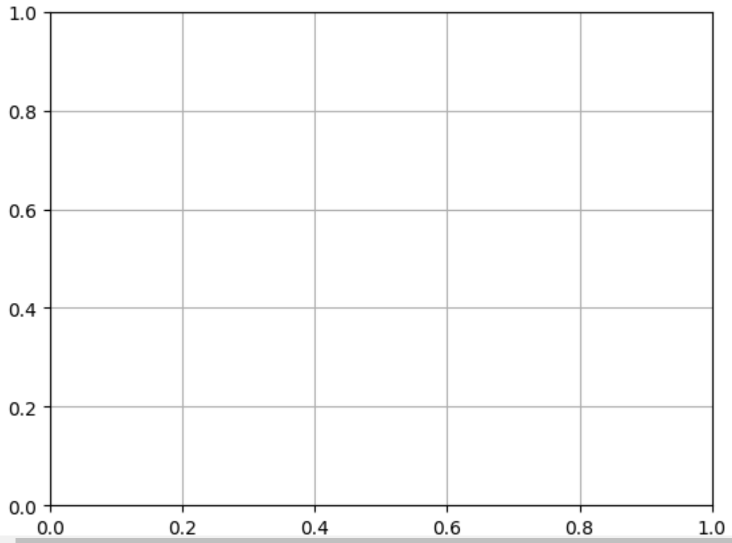
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-102-d5c7a14c8cd5> in <cell line: 1>()
----> 1 plt.plot(x,data,maker='0',color='blue',linestyle='_')

                                    3 frames
/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_base.py in _plot_args(self, tup, kwargs, return_kwargs,
ambiguous_fmt_datakey)
    502
    503         if x.shape[0] != y.shape[0]:
--> 504             raise ValueError(f"x and y must have same first dimension, but "
    505                              f"have shapes {x.shape} and {y.shape}")
    506         if x.ndim > 2 or y.ndim > 2:

ValueError: x and y must have same first dimension, but have shapes (10,) and (7,)
```

```
plt.title('line plot of data values')
plt.xlabel('index')
plt.ylabel('values')
```

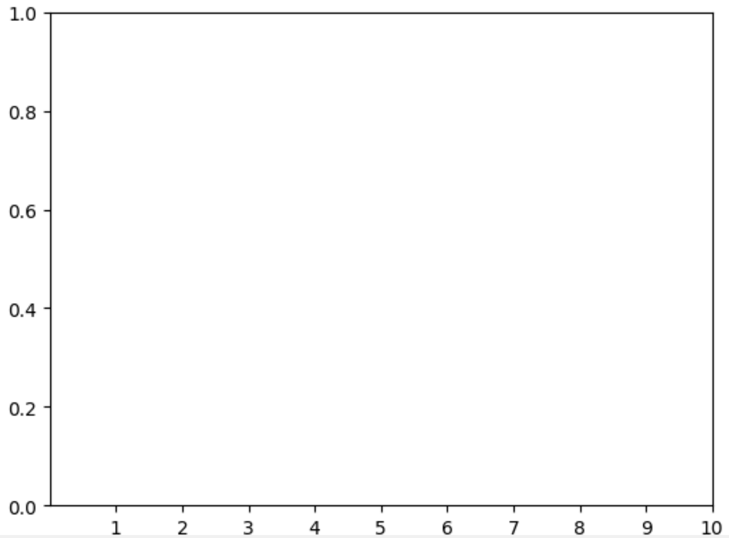
```
Text(0, 0.5, 'values')
```



```
plt.grid()
```

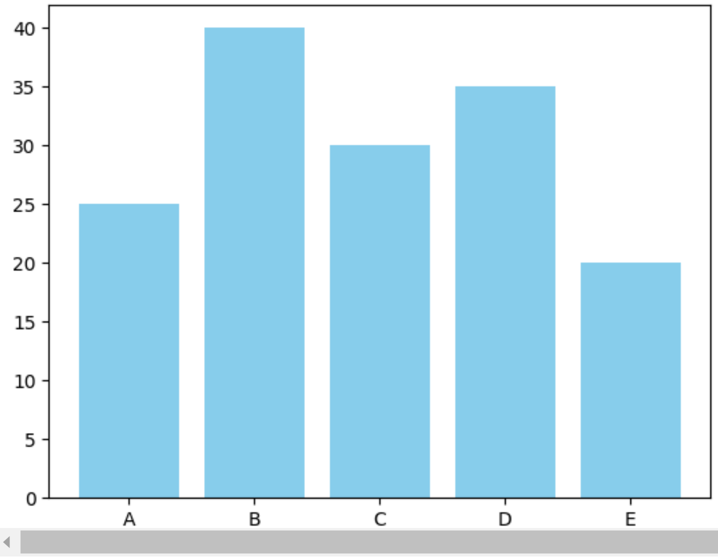


```
plt.xticks(x)
plt.show()
```



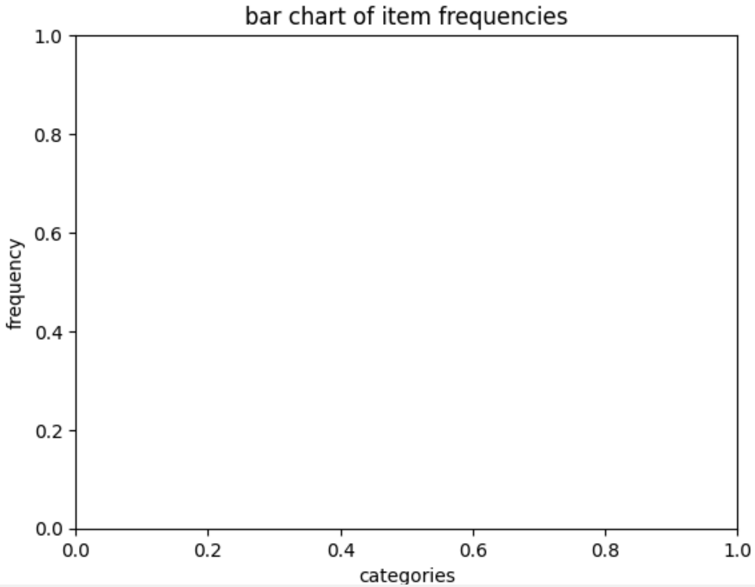
```
#question 3 data visualization assignment


import matplotlib.pyplot as plt



#given data
categories=['A','B','C','D','E']
values=[25,40,30,35,20]


#create bar chart
plt.bar(categories,values,color='skyblue')
```

```
<BarContainer object of 5 artists>
```

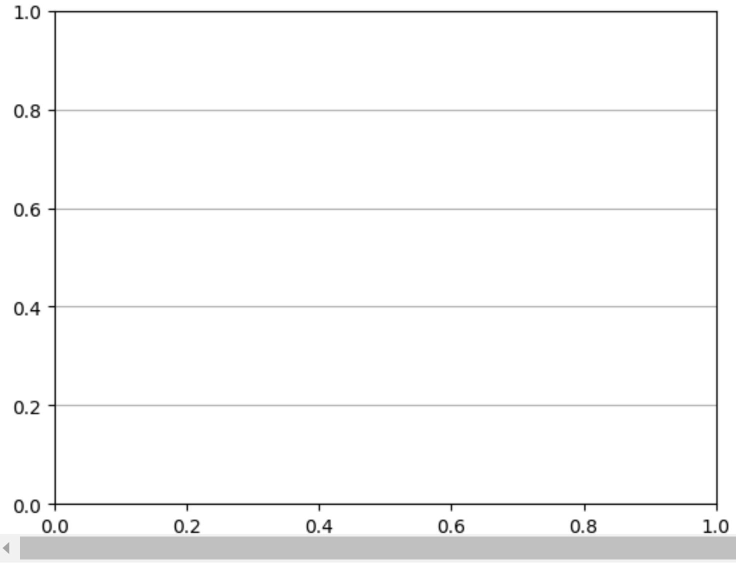


```
plt.title('bar chart of item frequencies')
plt.xlabel('categories')
plt.ylabel('frequency')
```

```
Text(0, 0.5, 'frequency')
```



```
#show grid
plt.grid(axis='y')
```



```
plt.show()
```

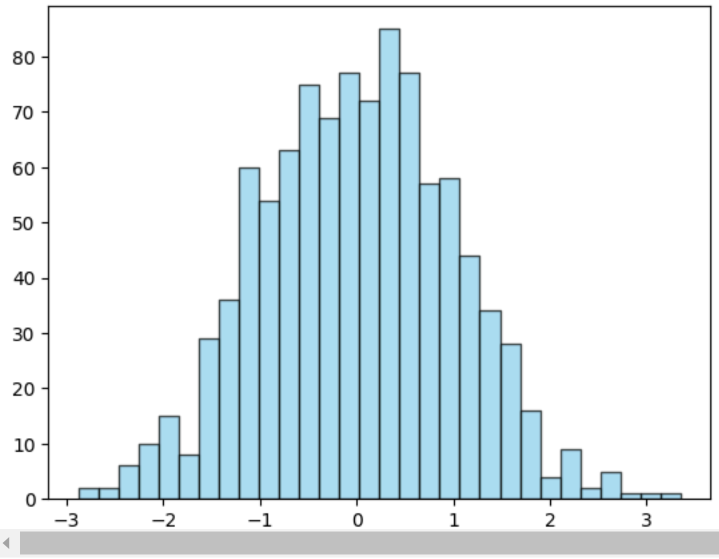
```
#qauestion no 4 data visualization assignment
```

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

```
data=np.random.normal(0,1,1000)
```

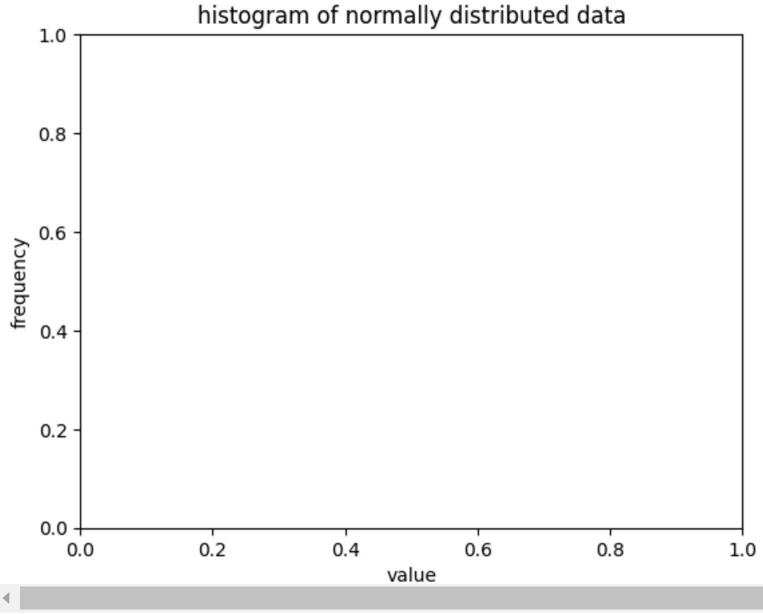
```
plt.hist(data,bins=30,color='skyblue',edgecolor='black',alpha=0.7)
```

```
(array([ 2.,  2.,  6., 10., 15.,  8., 29., 36., 60., 54., 63., 75., 69.,
        77., 72., 85., 77., 57., 58., 44., 34., 28., 16.,  4.,  9.,  2.,
         5.,  1.,  1.,  1.]),
 array([-2.87858888, -2.67085976, -2.46313064, -2.25540152, -2.0476724 ,
        -1.83994328, -1.63221416, -1.42448504, -1.21675592, -1.0090268 ,
        -0.80129768, -0.59356856, -0.38583944, -0.17811032,  0.0296188 ,
         0.23734792,  0.44507704,  0.65280616,  0.86053528,  1.06826441,
         1.27599353,  1.48372265,  1.69145177,  1.89918089,  2.10691001,
         2.31463913,  2.52236825,  2.73009737,  2.93782649,  3.14555561,
         3.35328473]),
 <BarContainer object of 30 artists>)
```

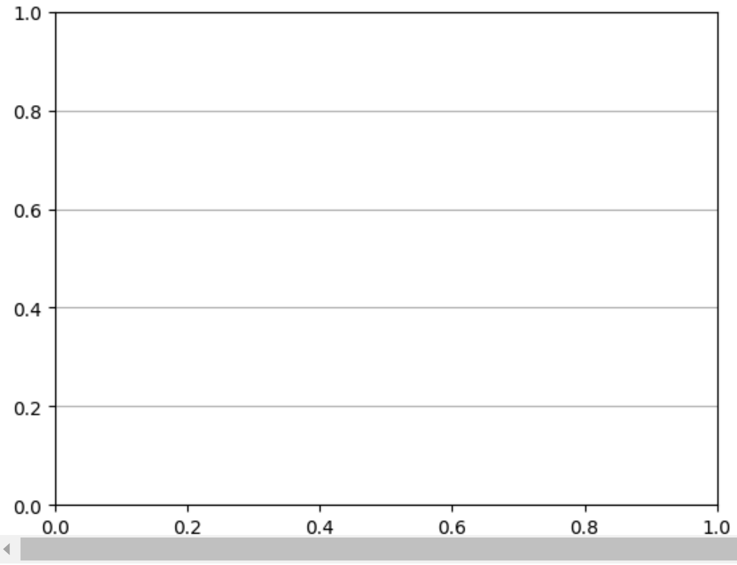


```
plt.title('histogram of normally distributed data')
plt.xlabel('value')
plt.ylabel('frequency')
```

```
Text(0, 0.5, 'frequency')
```

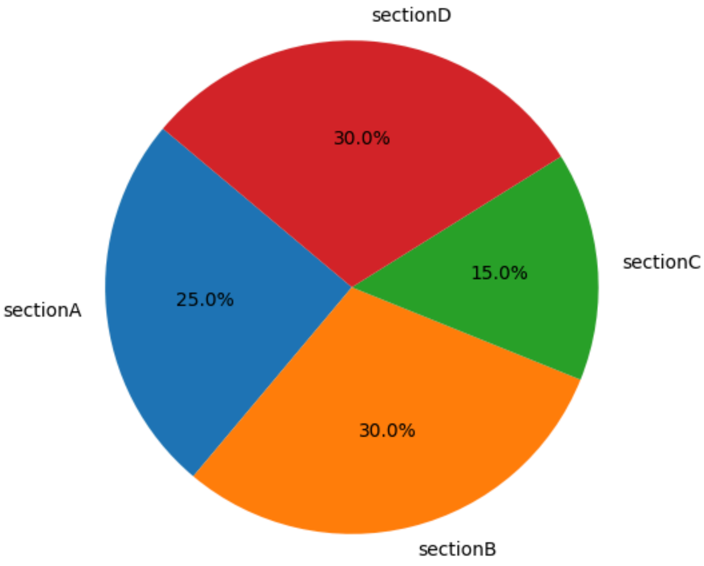


```
plt.grid(axis='y')
```



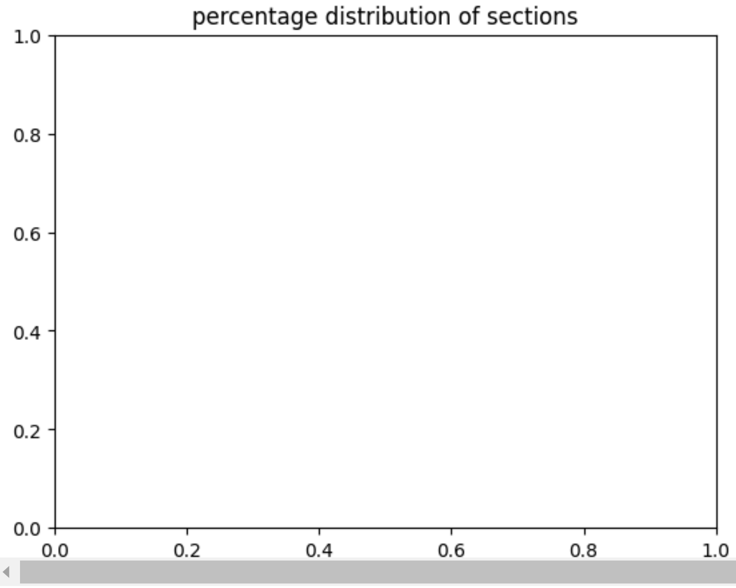
```
plt.show()


#question no 5 data visualization assignment MATPLOTLIB ASSIGNMENT


import matplotlib.pyplot as plt


#given data
sections=['sectionA','sectionB','sectionC','sectionD']
sizes=[25,30,15,30]


#creating pie chart
plt.figure(figsize=(8,6))
plt.pie(sizes,labels=sections,autopct='%1.1f%%',startangle=140)
```

```
([<matplotlib.patches.Wedge at 0x7b1fd06719f0>,
  <matplotlib.patches.Wedge at 0x7b1fd0673af0>,
  <matplotlib.patches.Wedge at 0x7b1fd0673130>,
  <matplotlib.patches.Wedge at 0x7b1fd03ef910>],
 [Text(-1.0958141679009201, -0.09587131702242374, 'sectionA'),
  Text(0.2661141251317168, -1.067325288937436, 'sectionB'),
  Text(1.095814158924795, 0.09587141961995002, 'sectionC'),
  Text(0.0767319567368486, 1.097320466780482, 'sectionD')],
 [Text(-0.5977168188550473, -0.05229344564859476, '25.0%'),
  Text(0.1451531591627546, -0.5821774303295105, '30.0%'),
  Text(0.5977168139589789, 0.05229350161088183, '15.0%'),
  Text(0.041853794583735594, 0.5985384364257172, '30.0%')])
```

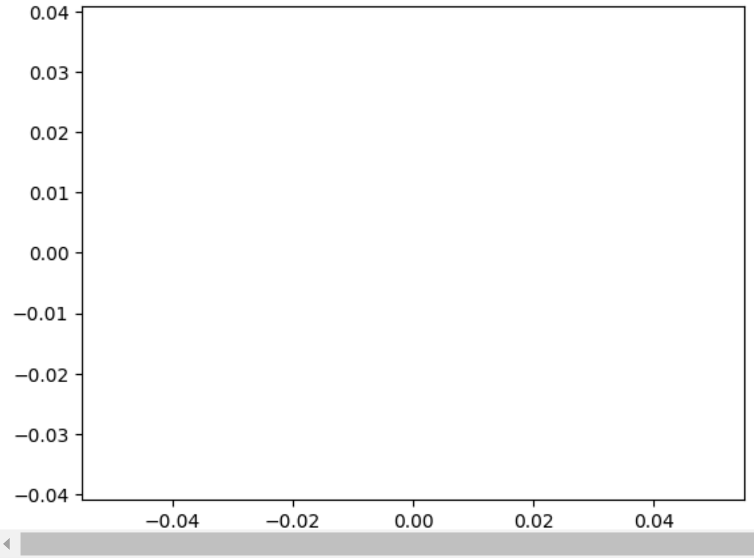


```
plt.title('percentage distribution of sections')
```

```
Text(0.5, 1.0, 'percentage distribution of sections')
```

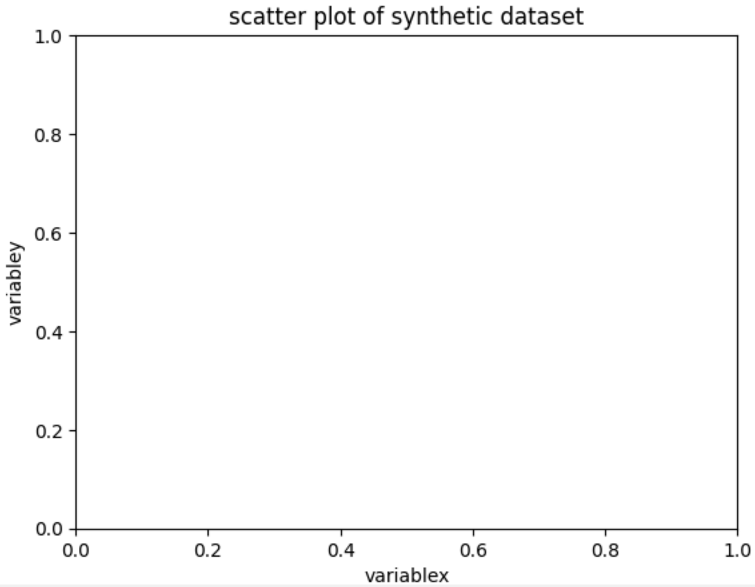


```
plt.axis('equal')
plt.show()
```

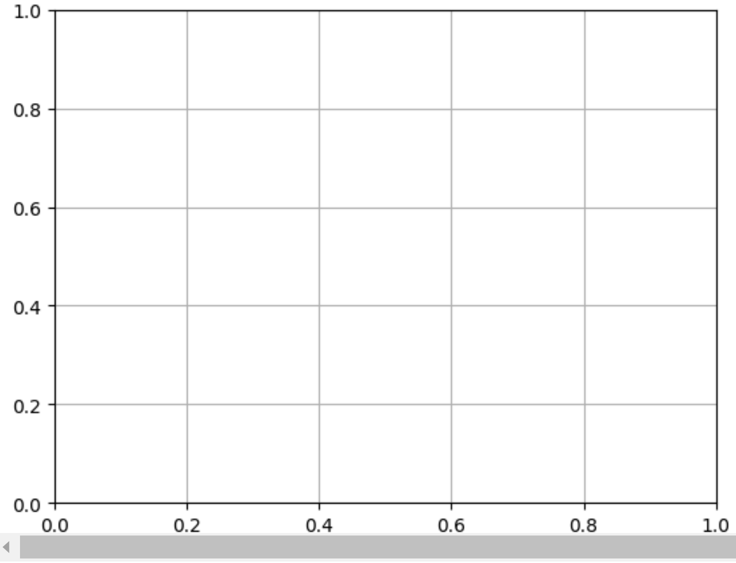


```
#SEABORN ASSIGNMENT STARTS HERE


#QUESTION NUM 1 SEABORN ASSIGNMENT
import numpy as np
import matplotlib.pyplot as plt


np.random.seed(42)


x=np.linspace(0,10,100)
y=2*x+np.random.normal(0,5,100)


plt.title('scatter plot of synthetic dataset')
plt.xlabel('variablex')
plt.ylabel('variabley')
```

```
Text(0, 0.5, 'variabley')
```

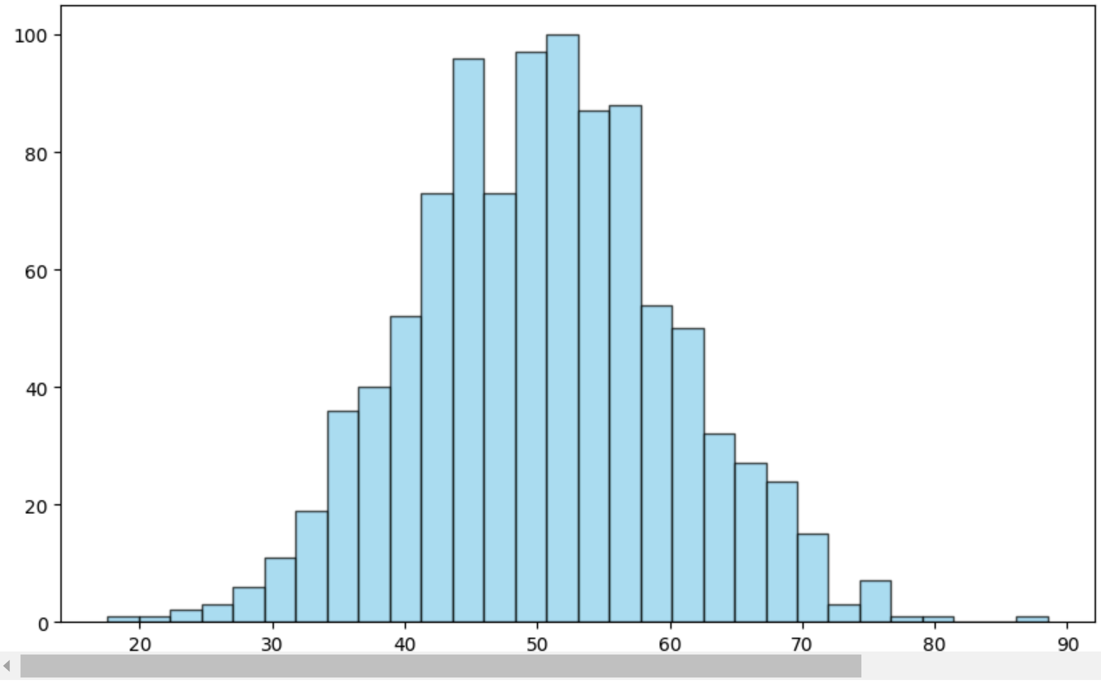


```
plt.grid()
```



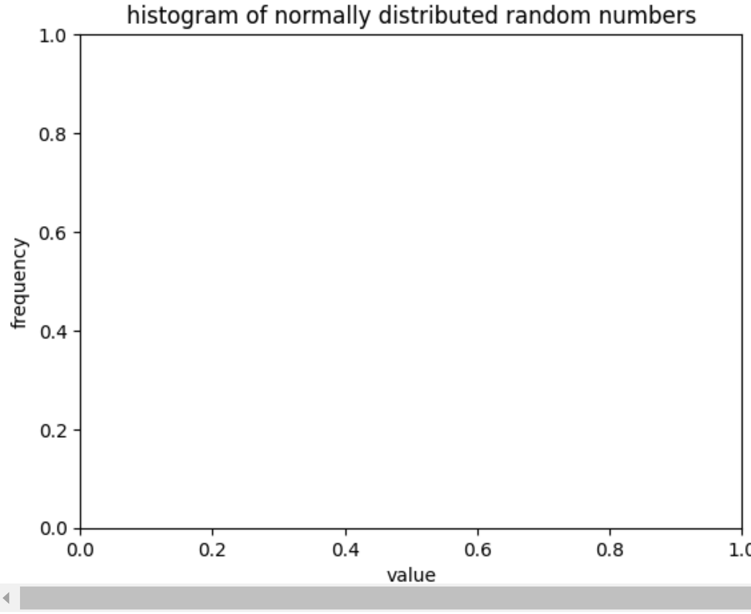
```
plt.show()


#QUESTION NO 2 SEABORN ASSIGNMENT
import numpy as np
import matplotlib.pyplot as plt


data=np.random.normal(loc=50,scale=10,size=1000)


plt.figure(figsize=(10,6))
plt.hist(data,bins=30,color='skyblue',edgecolor='black',alpha=0.7)
```

```
(array([  1.,   1.,   2.,   3.,   6.,  11.,  19.,  36.,  40.,  52.,  73.,
         96.,  73.,  97., 100.,  87.,  88.,  54.,  50.,  32.,  27.,  24.,
         15.,   3.,   7.,   1.,   1.,   0.,   0.,   1.]),
 array([17.5873266 , 19.95199288, 22.31665915, 24.68132543, 27.04599171,
        29.41065798, 31.77532426, 34.13999054, 36.50465681, 38.86932309,
        41.23398937, 43.59865565, 45.96332192, 48.3279882 , 50.69265448,
        53.05732075, 55.42198703, 57.78665331, 60.15131958, 62.51598586,
        64.88065214, 67.24531841, 69.60998469, 71.97465097, 74.33931725,
        76.70398352, 79.0686498 , 81.43331608, 83.79798235, 86.16264863,
        88.52731491]),
 <BarContainer object of 30 artists>)
```

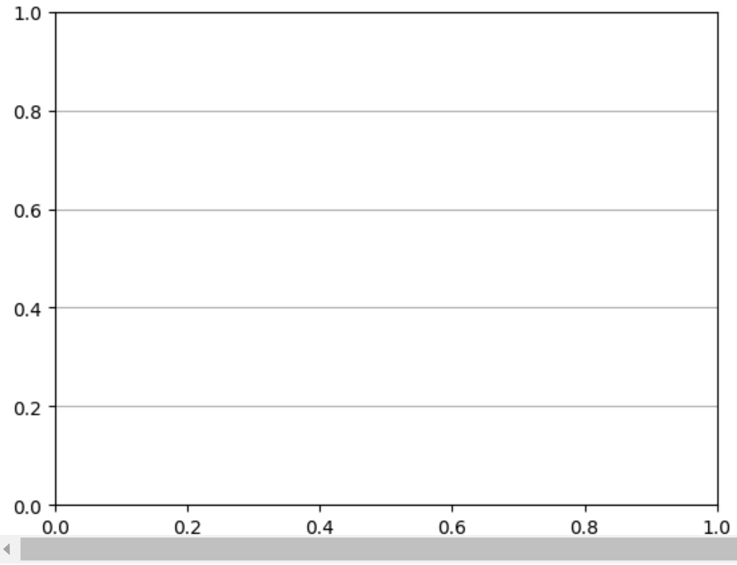


```
plt.title('histogram of normally distributed random numbers')
plt.xlabel('value')
plt.ylabel('frequency')
```

```
Text(0, 0.5, 'frequency')
```

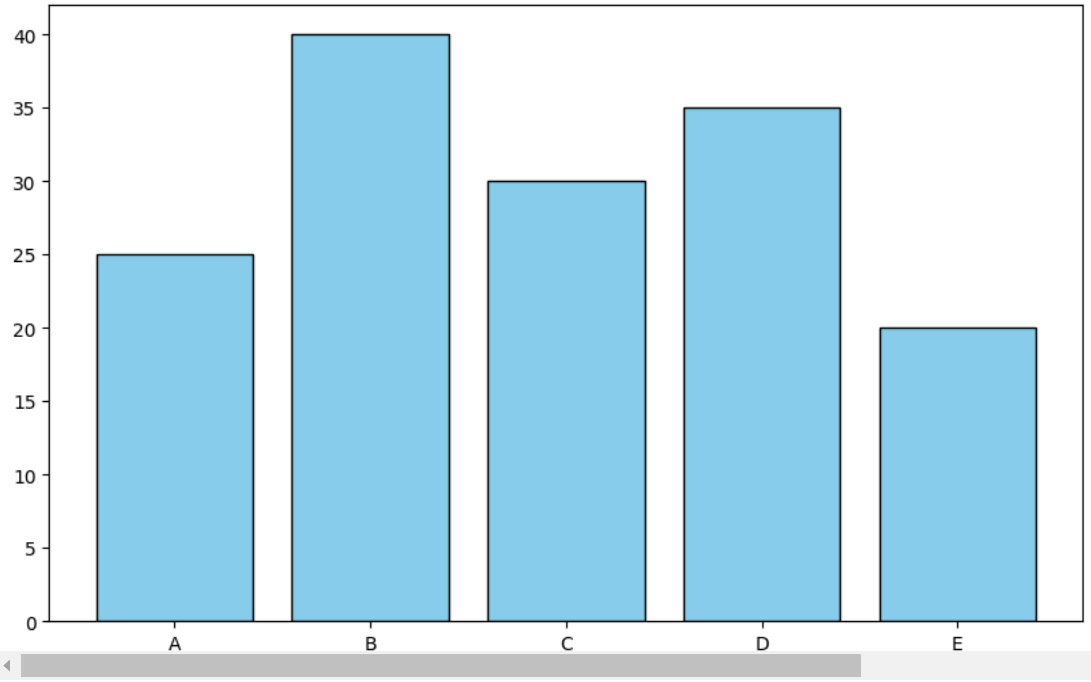


```
plt.grid(axis='y')
```



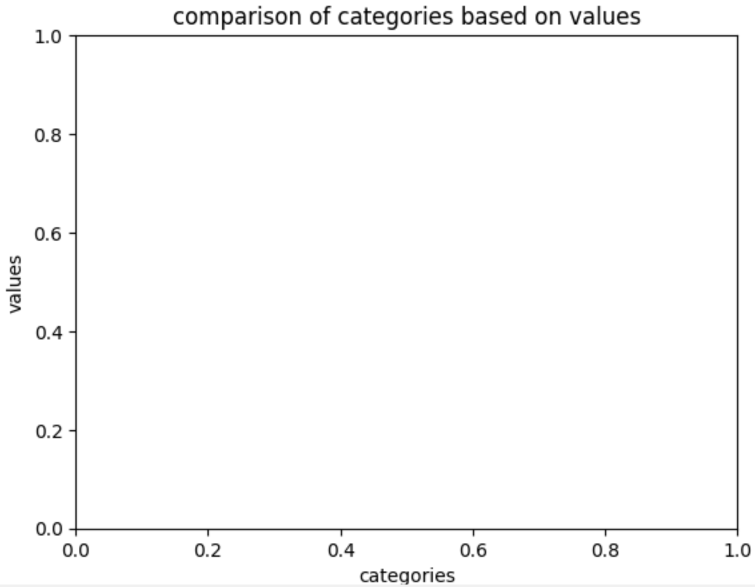
```
plt.show()


#QUESTION 3 SEABORN ASSIGNMENT
import matplotlib.pyplot as plt


categoriesa=['A','B','C','D','E']
valuesa=[25,30,15,30]


plt.figure(figsize=(10,6))
plt.bar(categoriesa,values,color='skyblue',edgecolor='black')
```

```
<BarContainer object of 5 artists>
```

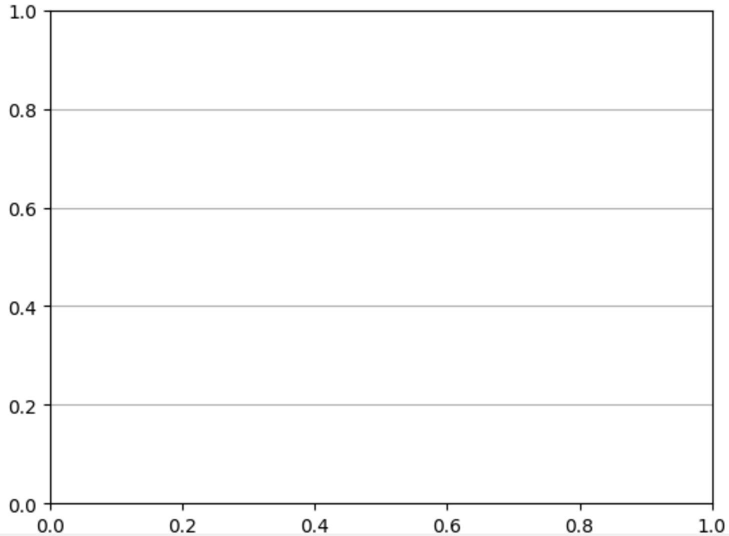


```
plt.title('comparison of categories based on values')
plt.xlabel('categories')
plt.ylabel('values')
```

```
Text(0, 0.5, 'values')
```



```
plt.grid(axis='y')
```



```
plt.show()


#question no 4 seaborn assignment


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


np.random.seed(42)


categories=['categoryA','categoryB','categoryC','categoryD']
data=
'category':np.random.choice.(categories,size=200),
'values':np.random.normal(loc=50,scale=10,size=200)
```

```
  File "<ipython-input-148-596504fc45ca>", line 2
    data=
         ^
SyntaxError: invalid syntax
```

```
#QUESTION NUMBER 5 SEABORN ASSIGNMENT
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

np.random.seed(42)


n_samples=1000
x=np.random.normal(0,1,n_samples)
y=2*x+np.random.normal(0,0.5,n_samples)
z=-1*x+np.random.normal(0,0.5,n_samples)


data= pd.dataframe({
    'feature 1':x,
    'feature 2':y,
    'feature 3':z
})
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-152-cae169351192> in <cell line: 1>()
----> 1 data= pd.dataframe({
      2     'feature 1':x,
      3     'feature 2':y,
      4     'feature 3':z
      5 })

AttributeError: module 'pandas' has no attribute 'dataframe'
```

```
corr_matrix=data.corr()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-153-dad23a0c43be> in <cell line: 1>()
----> 1 corr_matrix=data.corr()

AttributeError: 'numpy.ndarray' object has no attribute 'corr'
```

```
# plotly assignment
```

```
#question num 1 plotly assignment
```