

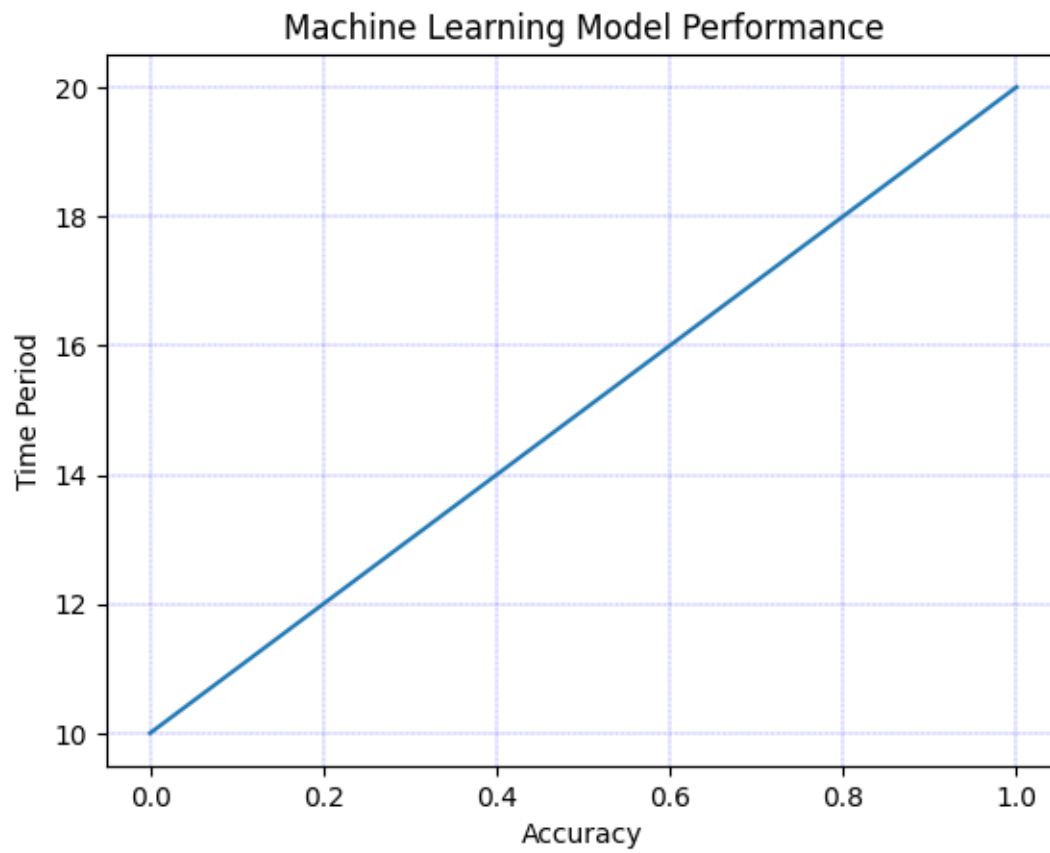
j6de7tbyb

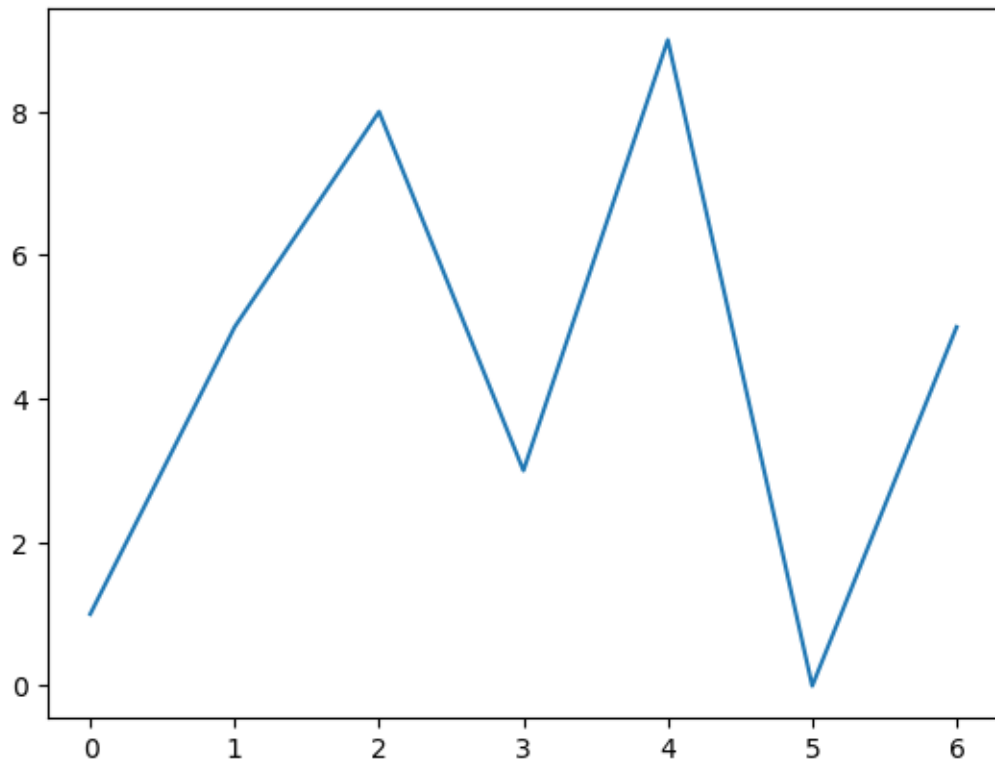
January 23, 2025

```
[ ]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

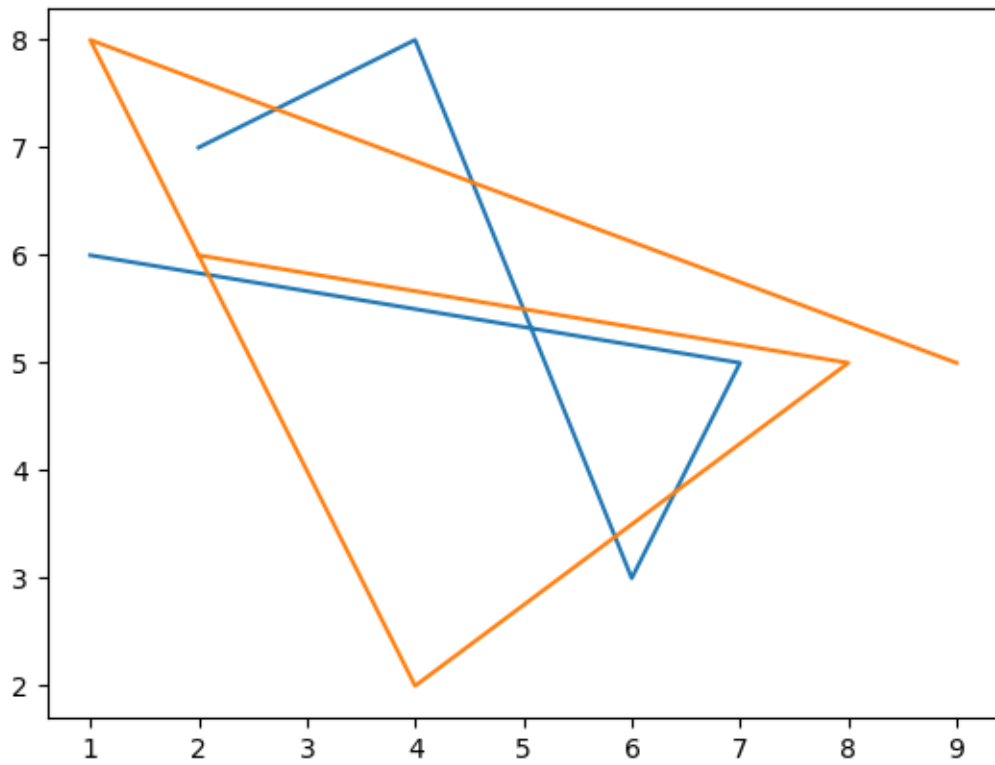
```
[ ]: data=[10,20]
plt.plot(data)
plt.xlabel("Accuracy")
plt.ylabel("Time Period")
plt.title("Machine Learning Model Performance")
plt.grid(color='blue',linewidth=0.2,linestyle="--")
plt.show()

data=[1,5,8,3,9,0,5]
plt.plot(data)
plt.show()
```

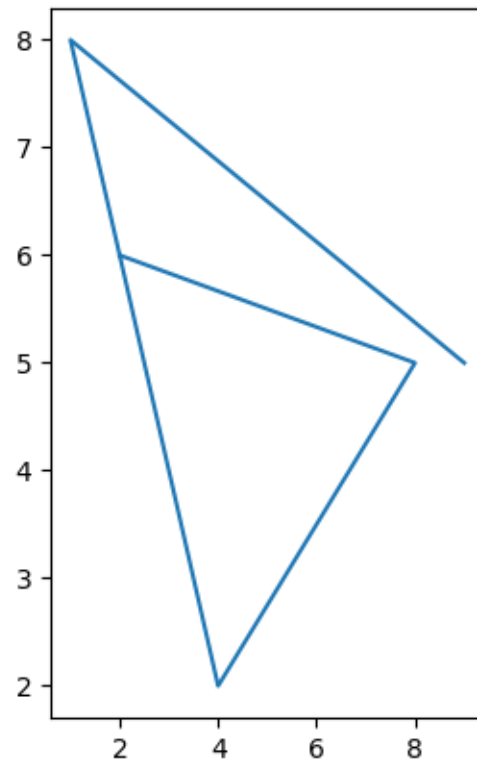
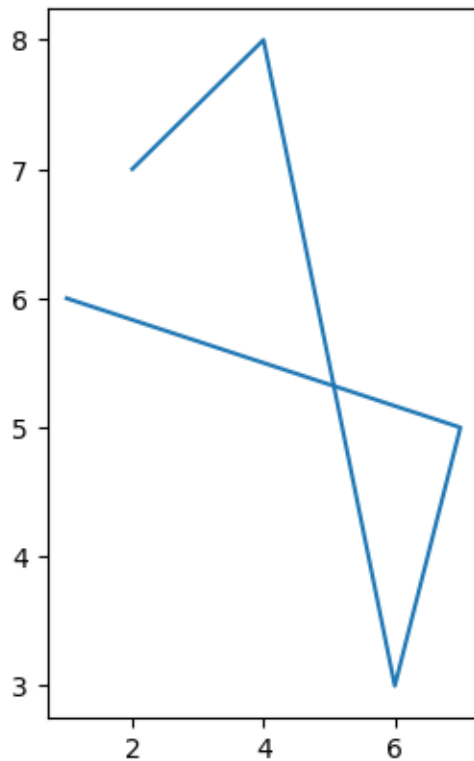




```
[ ]: #merged plots  
      #plot1  
      x=[2,4,6,7,1]  
      y=[7,8,3,5,6]  
      plt.plot(x,y)  
      #plot2  
      x=[9,1,4,8,2]  
      y=[5,8,2,5,6]  
      plt.plot(x,y)  
      plt.show()
```



```
[ ]: #sub plots
      #plot1
      plt.subplot(1,2,1)
      x=np.array([2,4,6,7,1])
      y=np.array([7,8,3,5,6])
      plt.plot(x,y)
      #plot2
      plt.subplot(1,2,2)
      x=np.array([9,1,4,8,2])
      y=np.array([5,8,2,5,6])
      plt.plot(x,y)
      plt.show()
```



```
[ ]: # Plot 1
plt.subplot(2, 2, 1) #2*2 figure and fig no. 1
x = np.array([2, 4, 6, 7, 1])
y = np.array([7, 8, 3, 5, 6])
plt.xlim(2,10) #limit the x-axis
plt.ylim(0,7)
plt.plot(x, y)

# Plot 2
plt.subplot(2, 2, 2)
x = np.array([9, 1, 4, 8, 2])
y = np.array([5, 8, 2, 5, 6])
plt.plot(x, y)

# Plot 3
plt.subplot(2, 2, 3)
x = np.array([6, 0, 1, 9, 3])
y = np.array([9, 3, 6, 1, 6])
plt.plot(x, y)

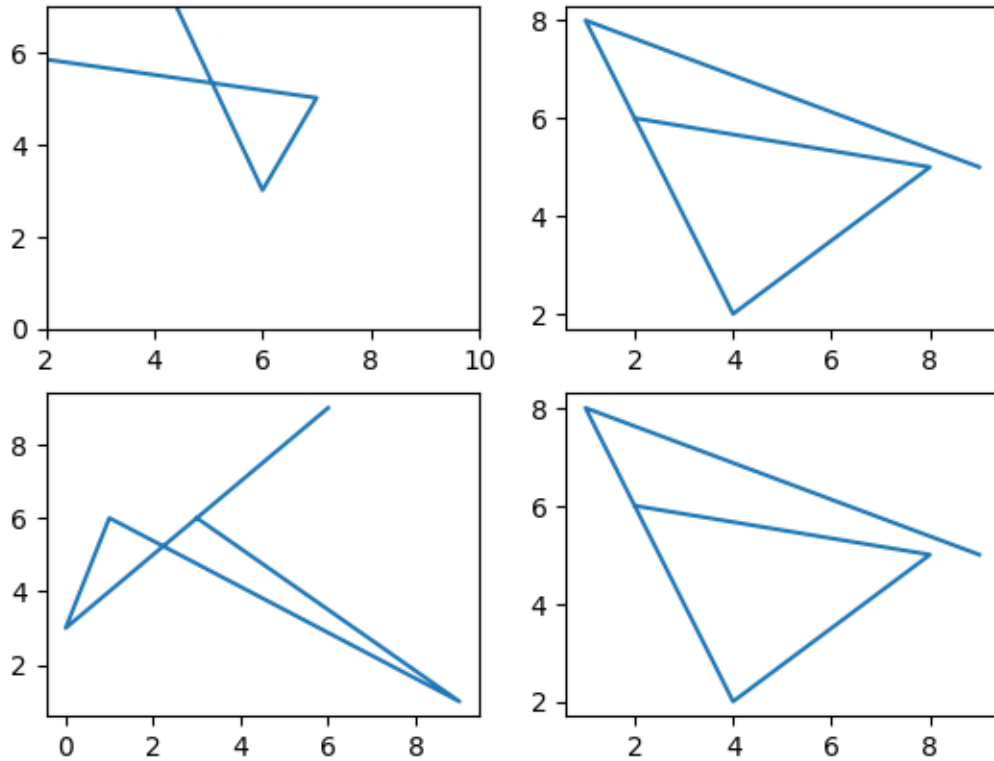
# Plot 4
plt.subplot(2, 2, 4)
```

```

x = np.array([9, 1, 4, 8, 2])
y = np.array([5, 8, 2, 5, 6])
plt.plot(x, y)

plt.show()

```



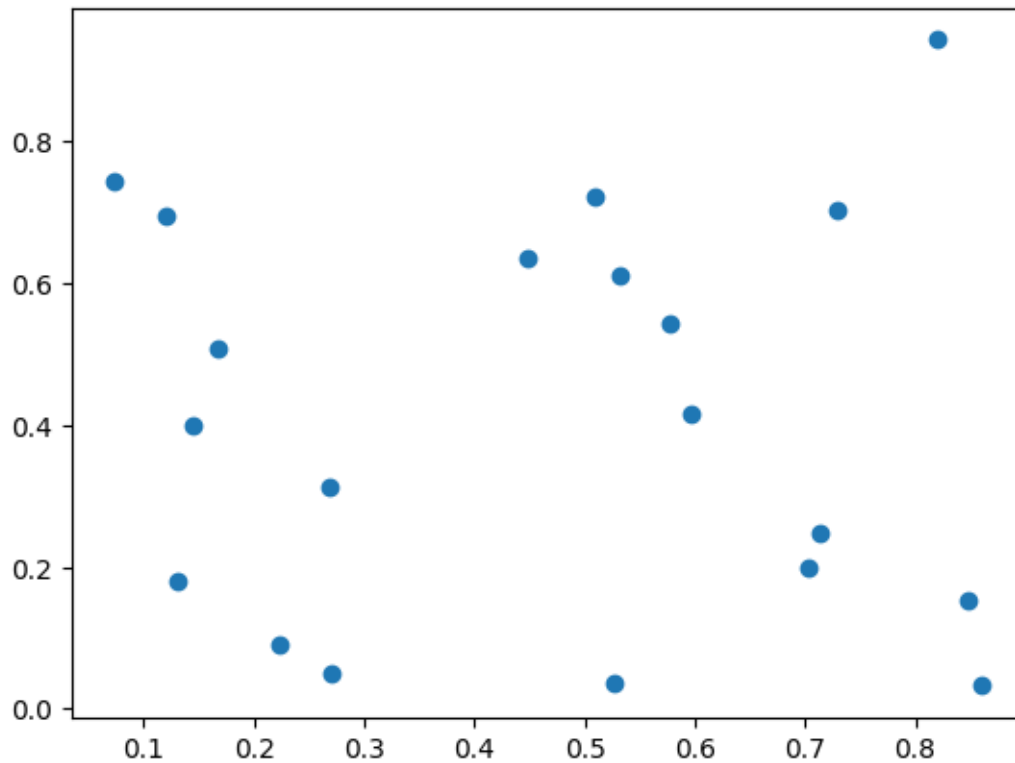
```

[ ]: x = np.array([0.26908831, 0.14426858, 0.50904045, 0.8603593, 0.72972161,
0.52629869, 0.27083531, 0.16748683, 0.59580926, 0.53141106,
0.22260323, 0.44845292, 0.57767903, 0.71262652, 0.07357533,
0.11958997, 0.70311875, 0.81980851, 0.84745334, 0.130776])

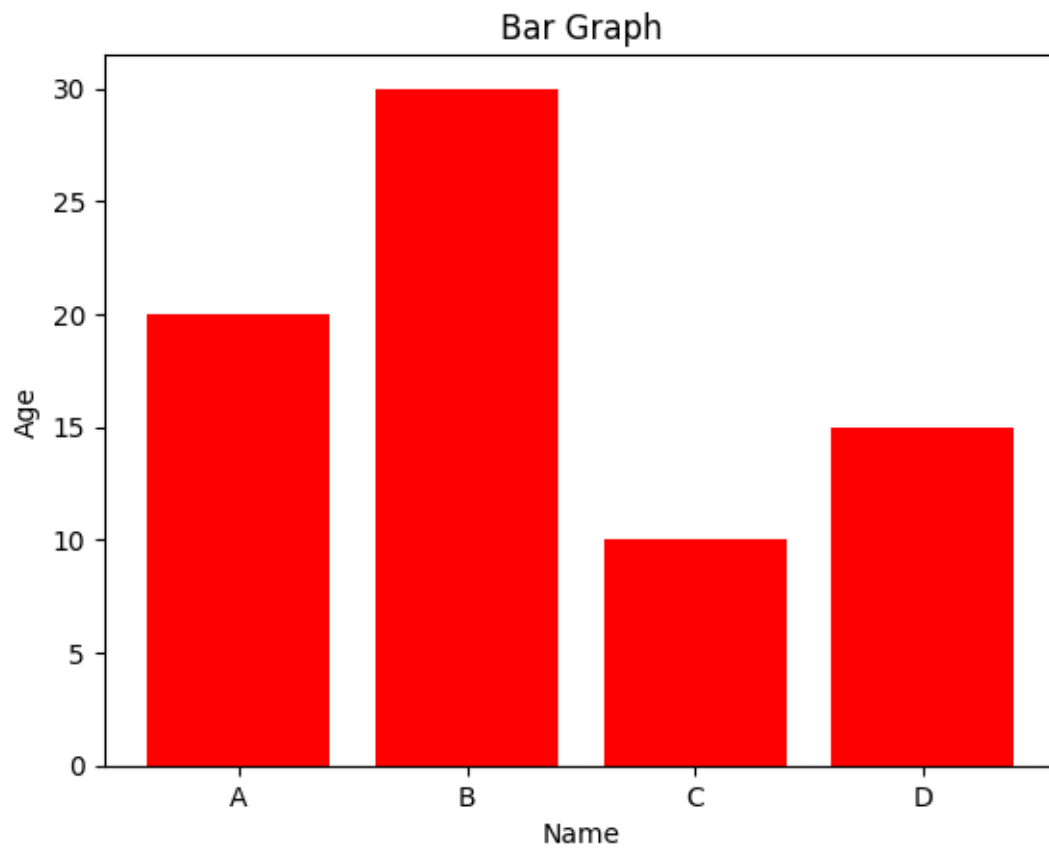
y = np.array([0.31405794, 0.4001774, 0.72212486, 0.03251845, 0.70225897,
0.03669069, 0.05121813, 0.50838174, 0.41513474, 0.61209124,
0.09179546, 0.63553532, 0.54260462, 0.24882753, 0.74348187,
0.69530417, 0.19834733, 0.94315613, 0.15405106, 0.18101433])

plt.scatter(x,y)
plt.show()

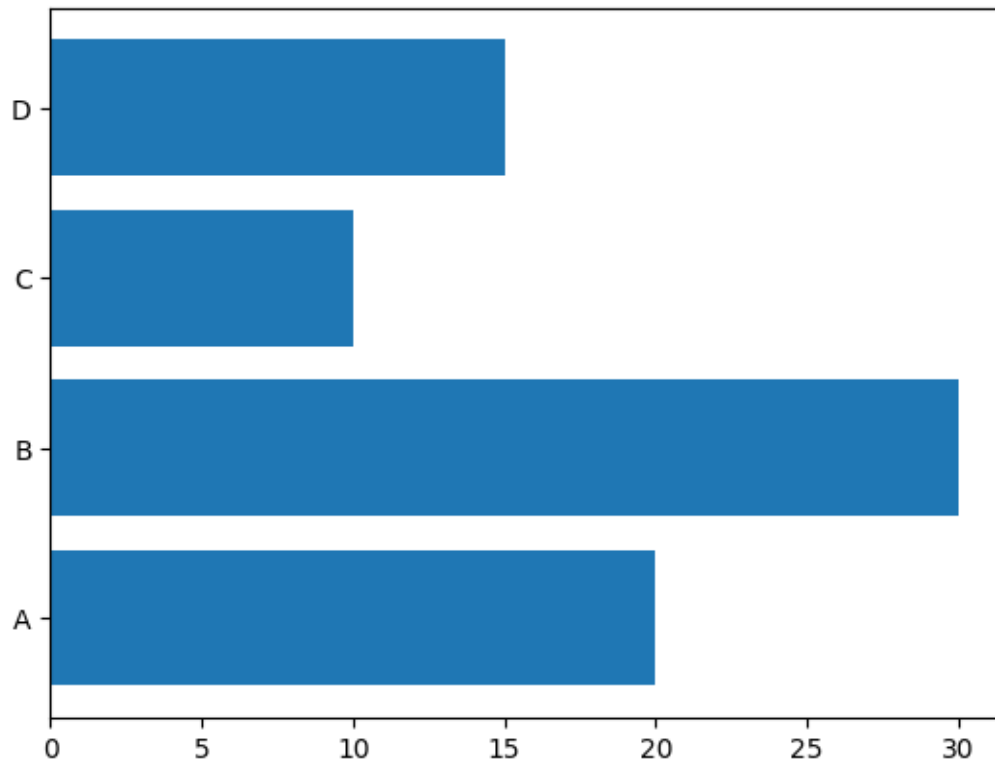
```



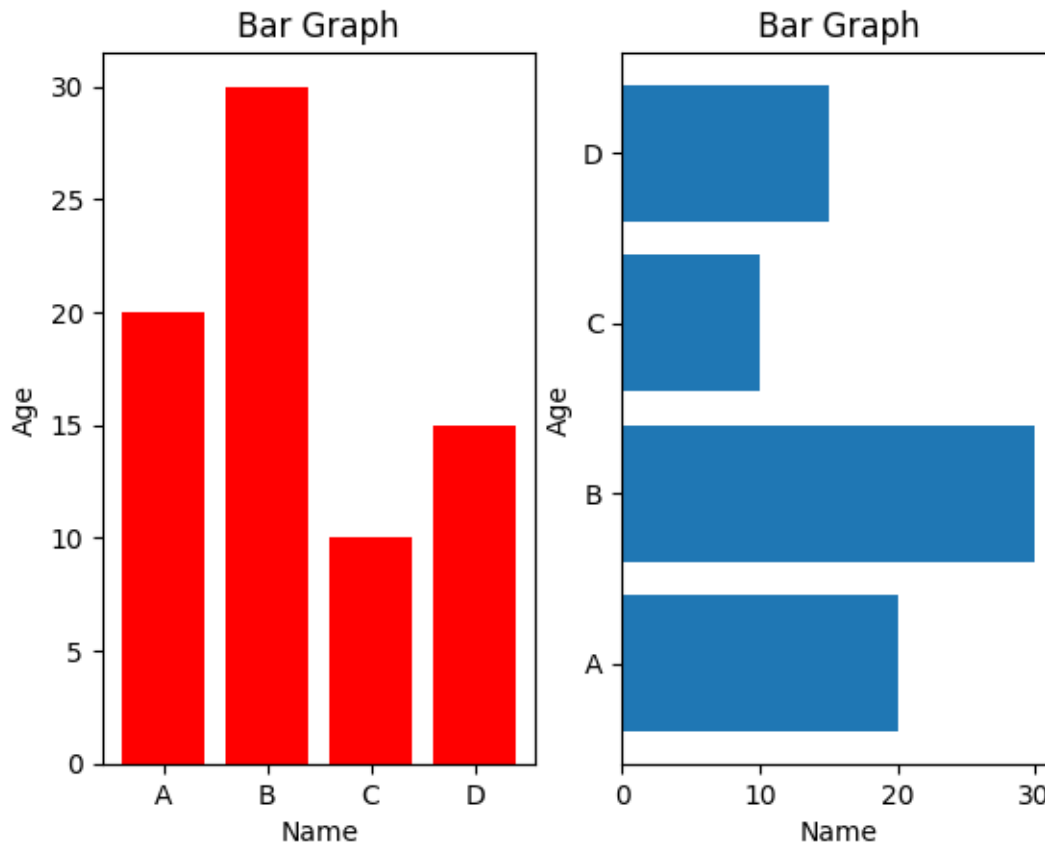
```
[ ]: x=np.array(['A','B','C','D'])
y=np.array([20,30,10,15])
plt.xlabel('Name')
plt.ylabel('Age')
plt.title('Bar Graph')
plt.bar(x,y,color='red')
plt.show() #creating a bar chart
plt.barh(x,y) #ceating a horizontal barchart
plt.show()
```







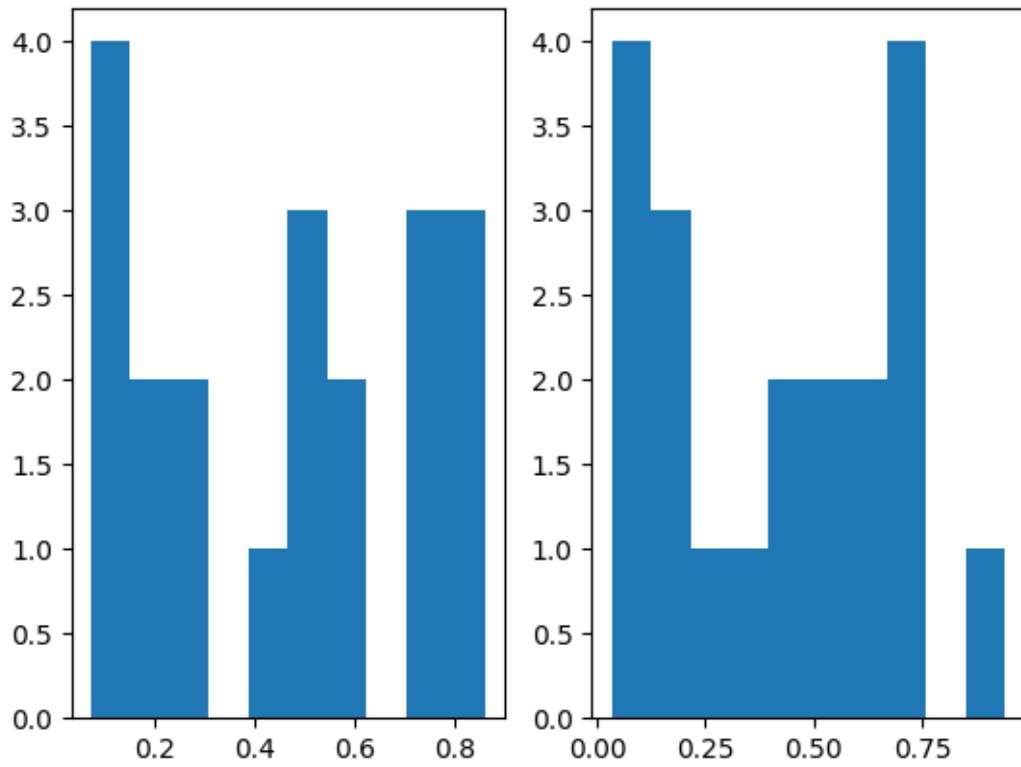
```
[ ]: x=np.array(['A','B','C','D'])
y=np.array([20,30,10,15])
plt.subplot(1,2,1)
plt.xlabel('Name')
plt.ylabel('Age')
plt.title('Bar Graph')
plt.bar(x,y,color='red')
plt.subplot(1,2,2)
plt.xlabel('Name')
plt.ylabel('Age')
plt.title('Bar Graph')
plt.barh(x,y) #ceating a horizontal barchart
plt.show()
```



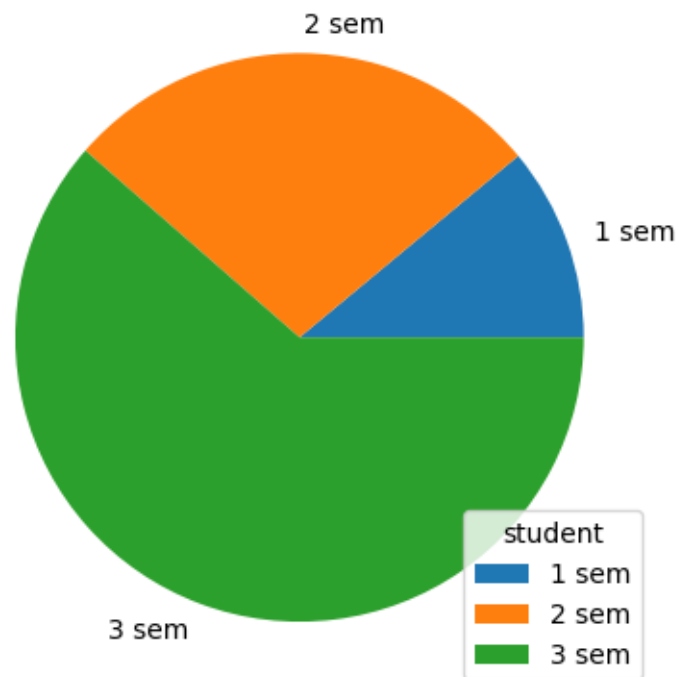
```
[ ]: x = np.array([0.26908831, 0.14426858, 0.50904045, 0.8603593, 0.72972161,
0.52629869, 0.27083531, 0.16748683, 0.59580926, 0.53141106,
0.22260323, 0.44845292, 0.57767903, 0.71262652, 0.07357533,
0.11958997, 0.70311875, 0.81980851, 0.84745334, 0.130776])

plt.subplot(1,2,1)
plt.hist(x)
y = np.array([0.31405794, 0.4001774, 0.72212486, 0.03251845, 0.70225897,
0.03669069, 0.05121813, 0.50838174, 0.41513474, 0.61209124,
0.09179546, 0.63553532, 0.54260462, 0.24882753, 0.74348187,
0.69530417, 0.19834733, 0.94315613, 0.15405106, 0.18101433])

plt.subplot(1,2,2)
plt.hist(y)
plt.show()
```



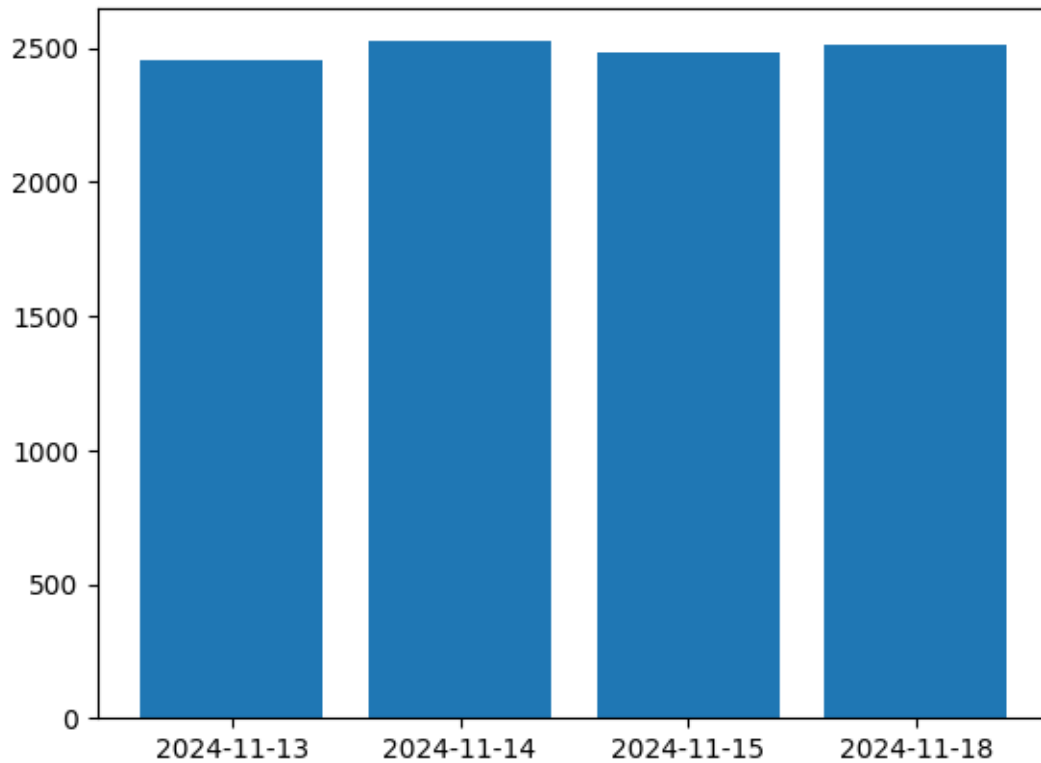
```
[ ]: labels=['1 sem','2 sem','3 sem']  
y=np.array([14,35,78])  
plt.pie(y,labels=labels)  
plt.legend(title='student')  
plt.show()
```



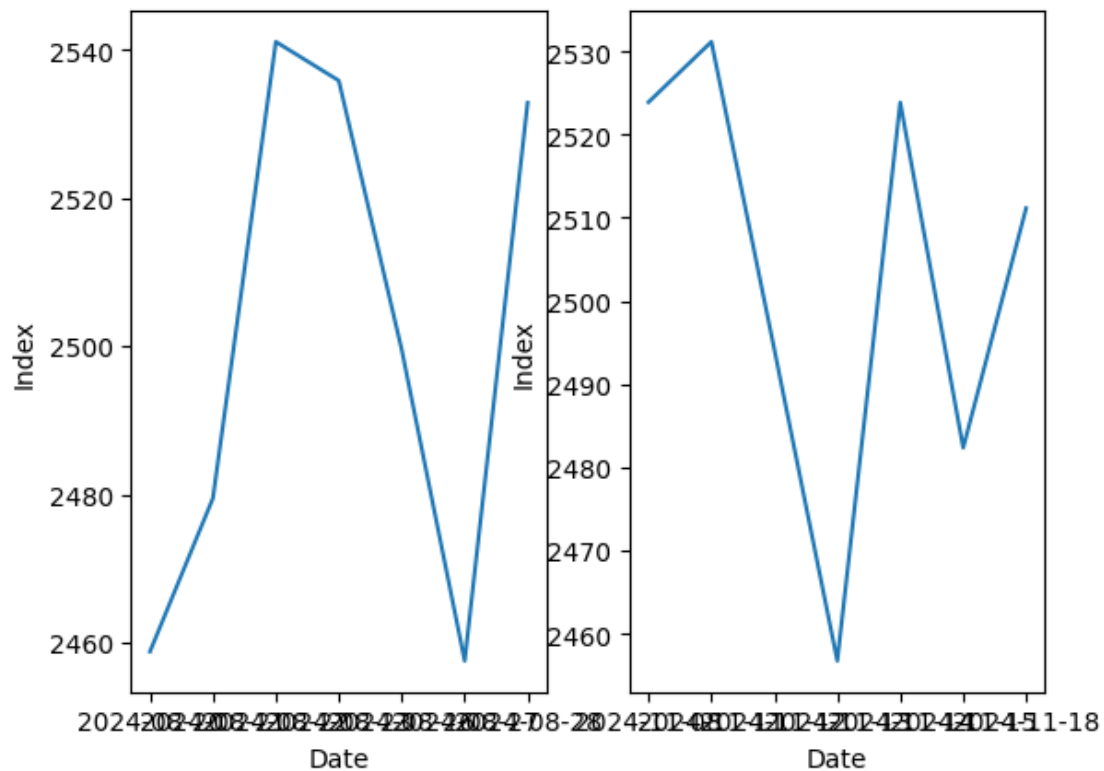
```
[ ]: #class assignment 11/19
df=pd.read_csv("nepse_last_90_days.csv")

[ ]: #last 4 days, sensitive index bar graph with labels and title
#first week data lineplot, lastweek data lineplot(subplot) with lables and title
last4=df.tail(4)
last4i=last4['NEPSE Index']
last4d=last4['Date']
plt.bar(last4d,last4i)

[ ]: <BarContainer object of 4 artists>
```

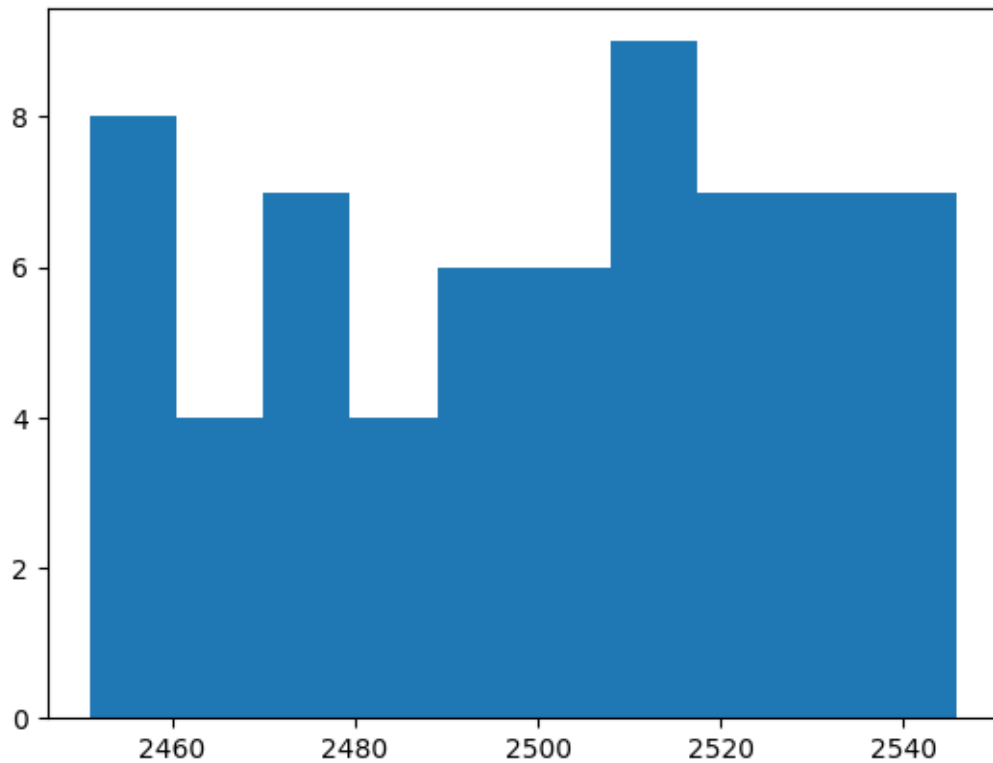


```
[ ]: first7=df.head(7)
first7d=first7['Date']
first7i=first7['NEPSE Index']
plt.subplot(1,2,1)
plt.xlabel('Date')
plt.ylabel('Index')
plt.plot(first7d,first7i)
last7=df.tail(7)
last7d=last7['Date']
last7i=last7['NEPSE Index']
plt.subplot(1,2,2)
plt.xlabel('Date')
plt.ylabel('Index')
plt.plot(last7d,last7i)
plt.show()
```



```
[ ]: val=df['NEPSE Index']
plt.hist(val)
```

```
[ ]: (array([8., 4., 7., 4., 6., 6., 9., 7., 7., 7.]),
array([2451.02 , 2460.498, 2469.976, 2479.454, 2488.932, 2498.41 ,
2507.888, 2517.366, 2526.844, 2536.322, 2545.8 ]),
<BarContainer object of 10 artists>)
```



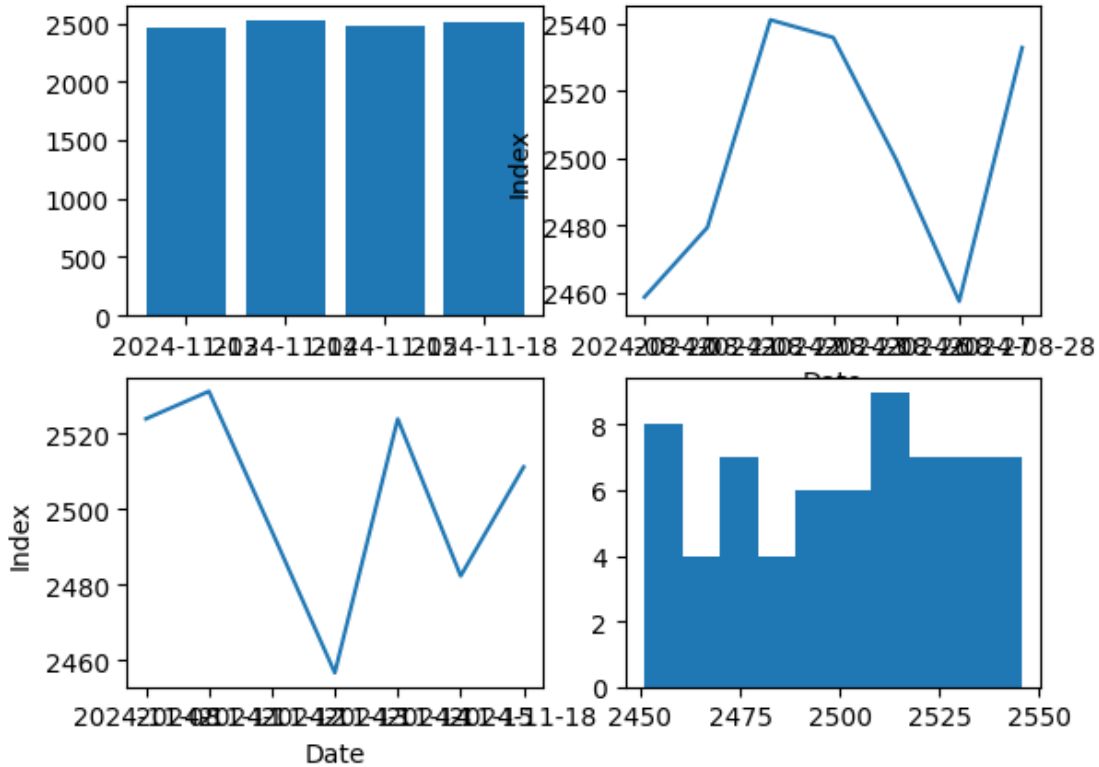
```
[ ]: last4=df.tail(4)
last4i=last4['NEPSE Index']
last4d=last4['Date']
plt.subplot(2,2,1)
plt.bar(last4d,last4i)

first7=df.head(7)
first7d=first7['Date']
first7i=first7['NEPSE Index']
plt.subplot(2,2,2)
plt.xlabel('Date')
plt.ylabel('Index')
plt.plot(first7d,first7i)
last7=df.tail(7)
last7d=last7['Date']
last7i=last7['NEPSE Index']
plt.subplot(2,2,3)
plt.xlabel('Date')
plt.ylabel('Index')
plt.plot(last7d,last7i)

val=df['NEPSE Index']
```

```
plt.subplot(2,2,4)
plt.hist(val)

plt.show()
```



```
[ ]: first7=df.head(7)
first7d=first7['Date']
x = pd.to_datetime(df['Date'])
x = first7d.dt.day_name()
first7i=first7['NEPSE Index']
plt.subplot(1,2,1)
plt.xlabel('Date')
plt.ylabel('Index')
plt.plot(x,first7i)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-88-7f9a6bb2e527> in <cell line: 9>()
      7 plt.xlabel('Date')
      8 plt.ylabel('Index')
----> 9 plt.plot(x,first7i)
```



```

/usr/local/lib/python3.10/dist-packages/matplotlib/pyplot.py in plot(scalex,
↳ scalex, data, *args, **kwargs)
    3576     **kwargs,
    3577 ) -> list[Line2D]:
-> 3578     return gca().plot(

    3579         *args,
    3580         scalex=scalex,

/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_axes.py in plot(self,
↳ scalex, scaley, data, *args, **kwargs)
    1719         """
    1720         kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1721         lines = [*self._get_lines(self, *args, data=data, **kwargs)]
    1722         for line in lines:
    1723             self.add_line(line)

/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_base.py in
↳ __call__(self, axes, data, *args, **kwargs)
    301             this += args[0],
    302             args = args[1:]
-> 303             yield from self._plot_args(

    304                 axes, this, kwargs,
↳ ambiguous_fmt_datakey=ambiguous_fmt_datakey)
    305

/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_base.py in
↳ _plot_args(self, axes, tup, kwargs, return_kwargs, ambiguous_fmt_datakey)
    497
    498         if x.shape[0] != y.shape[0]:
-> 499             raise ValueError(f"x and y must have same first dimension,
↳ but "

    500                                     f"have shapes {x.shape} and {y.shape}")
    501         if x.ndim > 2 or y.ndim > 2:

ValueError: x and y must have same first dimension, but have shapes (65,) and
↳ (7,)

```

