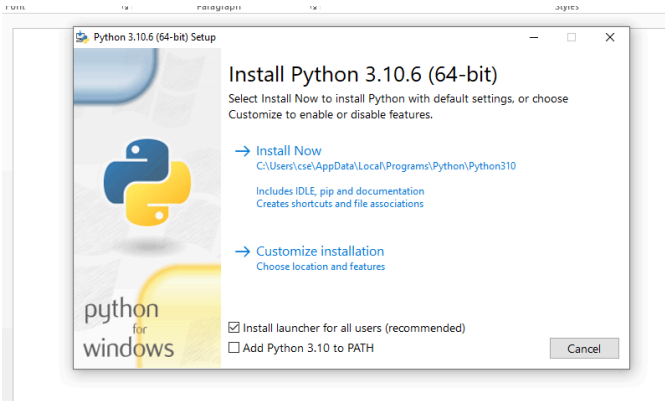
Using Numpy,Pandas and Matplotlib libraries

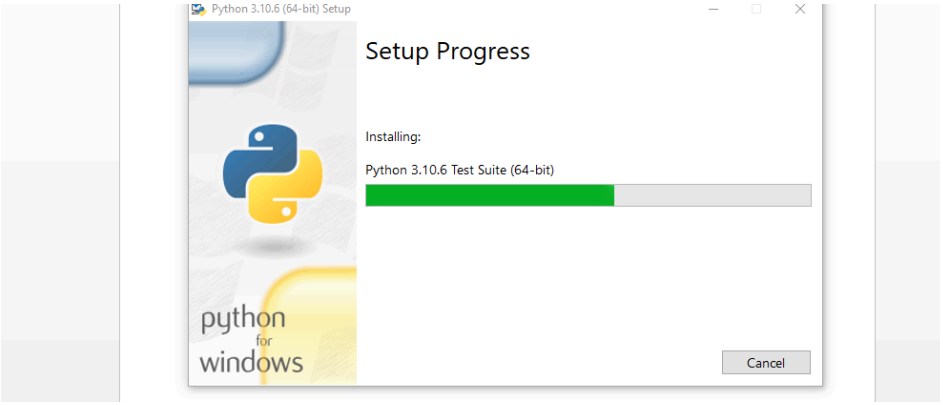
Python Installation:

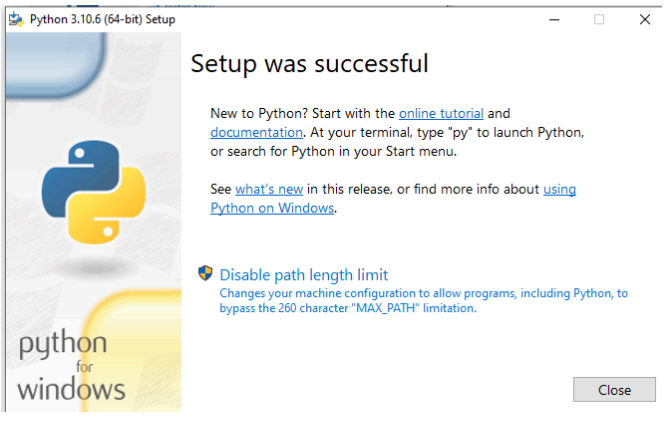
Go to https://www.python.org/downloads/

Select latest version of python and download it.

Run the .exe has Adminstrator and select Install Now option.







2.Downloading Numpy,pandas,matplotlib.

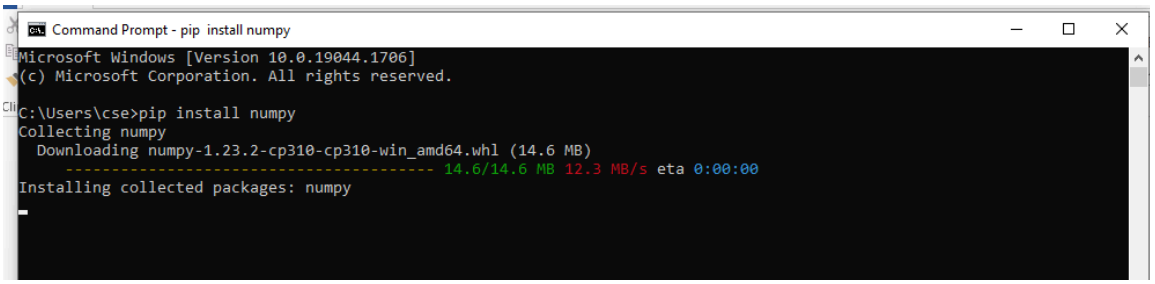
1.Numpy:

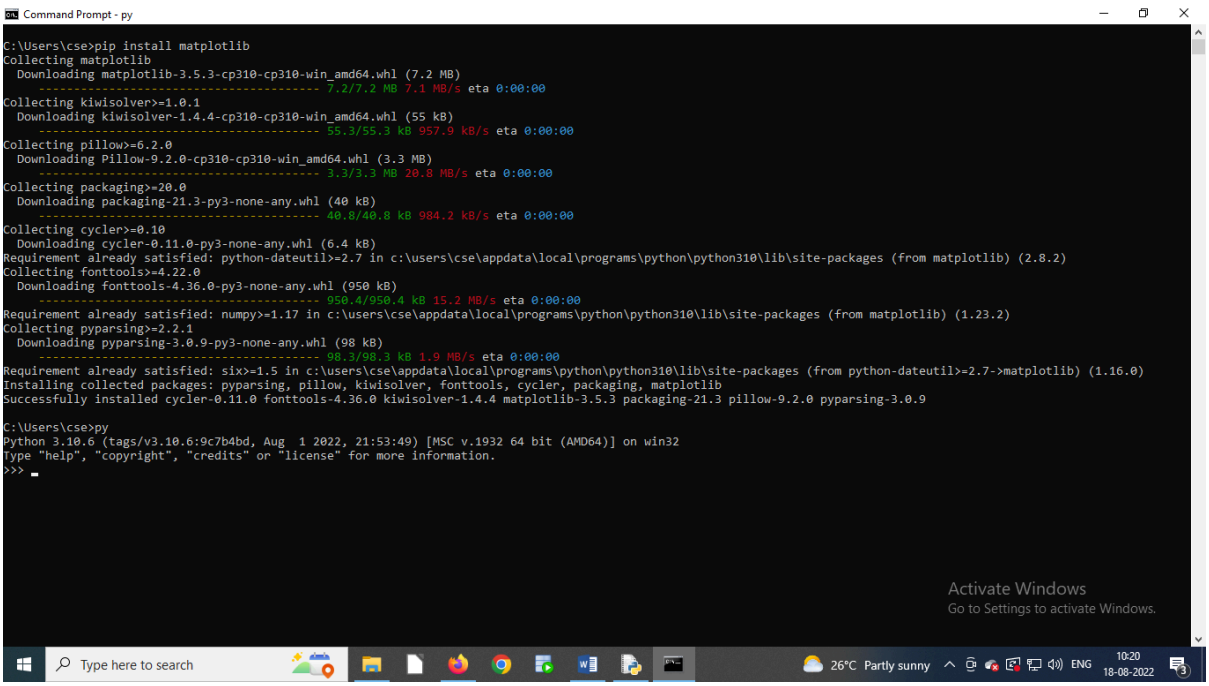
To download numpy

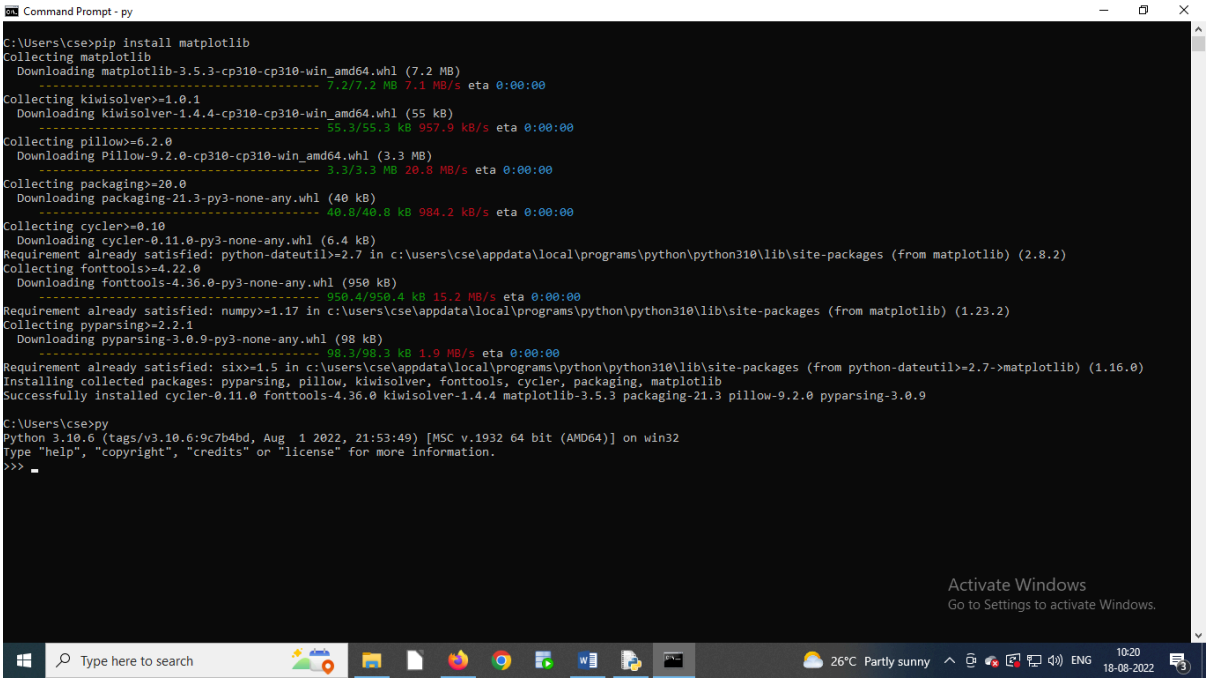
pip install numpy

pip install pandas

pip install matplotlib







In [1]:
```
import numpy as np
a=(np.arange(8)*5).reshape(2,4)
print(a)
for i in np.nditer(a):
    print(i,end=" ")
```

```
[[ 0  5 10 15]
 [20 25 30 35]]
0 5 10 15 20 25 30 35
```

In [6]:
```
import numpy as np
a=np.array([[[2,3],[4,5]],
            [[6,7],[8,9]]])
b=np.random.rand(5)
#writing to file
file=open("a","wb")
np.save(file,a)
file.close
file=open("b","wb")
np.save(file,b)
file.close
#reading from file
file=open("a","rb")
arr=np.load(file)
print(arr)
file=open("b","rb")
br=np.load(file)
print(b)
```

```
[[[2 3]
  [4 5]]

 [[6 7]
  [8 9]]]
[0.37401795 0.5182383  0.47456251 0.3130446  0.69531923]
```

In [3]:
```
#using Random.rand to create 2-D & 3-D arrays
import numpy as np
one=np.random.rand(10)
two=np.random.rand(2,6)
three=np.random.rand(4,2,2)
print('1-d array:')
print(one)
print('2-d array with 6 elements:')
print(two)
print('3-d array with 4 2-d array having elements')
print(three)
```

```
1-d array:
[0.17499826 0.08477758 0.82574592 0.49544258 0.18568158 0.96301879
 0.47673319 0.88486393 0.56169756 0.30366595]
2-d array with 6 elements:
[[0.5583047  0.75535492 0.35464548 0.32313653 0.43046965 0.54100808]
 [0.69194475 0.56264919 0.28121458 0.71192487 0.90754478 0.3696306 ]]
3-d array with 4 2-d array having elements
[[[0.09961528 0.44354215]
  [0.47753741 0.74993621]]

 [[0.18682951 0.6770017 ]
  [0.57536301 0.53181343]]


       [[0.98659711 0.83887312]
        [0.32676059 0.04352788]]

       [[0.80991416 0.95164097]
        [0.93357962 0.09111596]]]
```

In [4]:
```
import numpy as np
a=np.array([3,4,5,6])
b=np.arange(10,100,10)
c=np.linspace(1,10,5)
#d=np.random.rand(1)
print("numpy library:")
print("using array:",a)
shape=a.shape
print(shape)
print('using arange:',b)
shape=b.shape
print(shape)
print('using linspace:',c)
shape=c.shape
print(shape)
```

```
numpy library:
using array: [3 4 5 6]
(4,)
using arange: [10 20 30 40 50 60 70 80 90]
(9,)
using linspace: [ 1.    3.25  5.5   7.75 10.  ]
(5,)
```

In [7]:
```
import numpy as np
import pandas as pd
s=pd.Series([1,2,3,"pandas","numpy",np.nan,9,7])
print(s)
print(s[4])
mydic={'student name':['sam','john','james','rose'],
    'roll':[1,3,2,4]
       }
dataframe=pd.DataFrame(mydic)
print(dataframe)
```

```
0         1
1         2
2         3
3    pandas
4     numpy
5       NaN
6         9
7         7
dtype: object
numpy
  student name  roll
0          sam     1
1         john     3
2        james     2
3         rose     4
```

2.Aim: To find mean ,median ,variance and standard deviation using numpy.

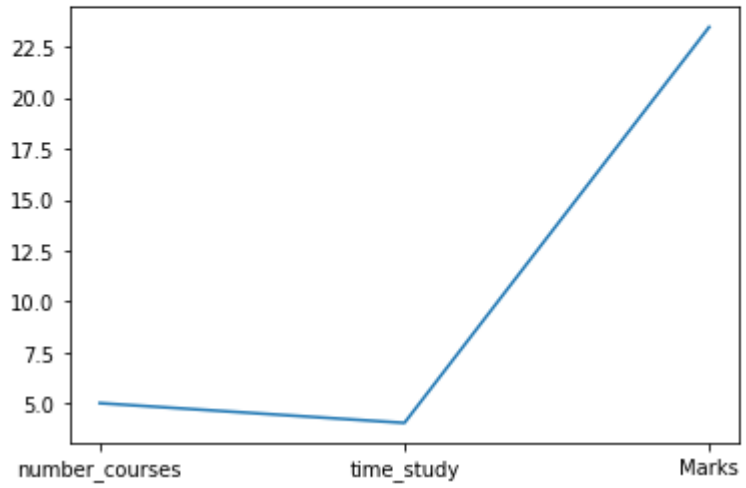
Mean:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from scipy import stats
data=pd.read_csv("C:/Users/Admin/Downloads/Student_Marks.csv").head(50)
x=np.mean(data)
print(x)
plt.plot(x)
```

output:

```
number_courses     5.0200
time_study         4.0499
Marks             23.4799
dtype: float64

[<matplotlib.lines.Line2D at 0x1ba833ffe80>]
```



Median:

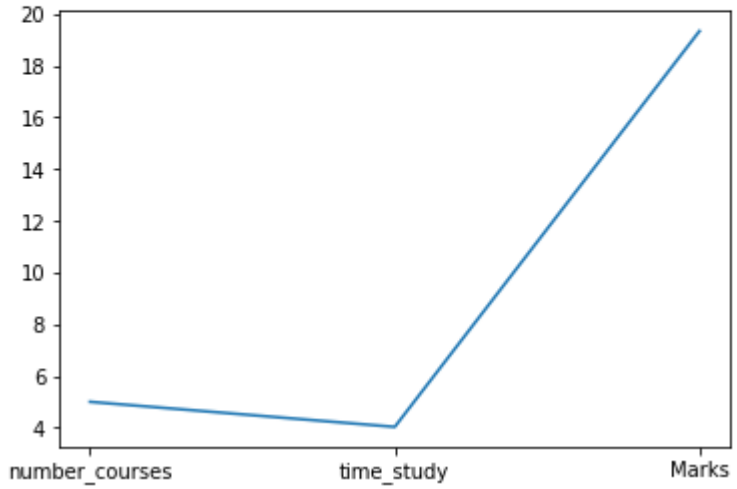
```
m=data.median()
print(m)
```

plt.plot(m)

Output:

```
number_courses     5.000
time_study         4.030
Marks             19.334
dtype: float64
```

```
[<matplotlib.lines.Line2D at 0x1ba83503a30>]
```



Mode:

mo=data.mode()

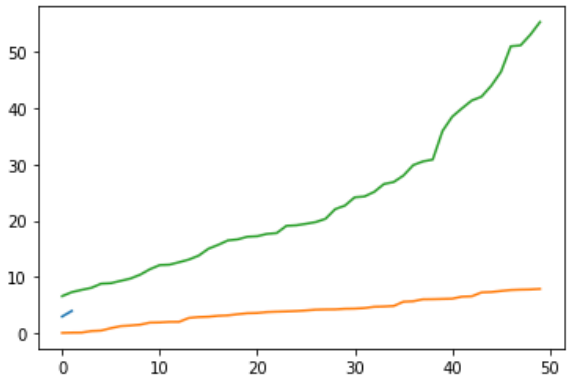
print(mo)

plt.plot(mo)

Output:

	number_courses	time_study	Marks
0	3.0	0.096	6.623
1	4.0	0.140	7.336
2	NaN	0.156	7.734
3	NaN	0.423	8.100
4	NaN	0.508	8.837
5	NaN	0.932	8.924
6	NaN	1.299	9.333
7	NaN	1.407	9.742
8	NaN	1.557	10.429
9	NaN	1.923	11.397
10	NaN	1.954	12.132
11	NaN	2.051	12.209
12	NaN	2.061	12.647
13	NaN	2.754	13.119
14	NaN	2.908	13.811
15	NaN	2.966	15.038
16	NaN	3.133	15.725
17	NaN	3.211	16.517
18	NaN	3.413	16.703
19	NaN	3.591	17.171
20	NaN	3.635	17.264
21	NaN	3.797	17.672
22	NaN	3.864	17.822
23	NaN	3.913	19.106
24	NaN	3.977	19.202
25	NaN	4.083	19.466

```
Out[14]: [<matplotlib.lines.Line2D at 0x1ba8356e6d0>,
          <matplotlib.lines.Line2D at 0x1ba8356e700>,
          <matplotlib.lines.Line2D at 0x1ba8356e820>]
```



variance:

v=np.var(data)

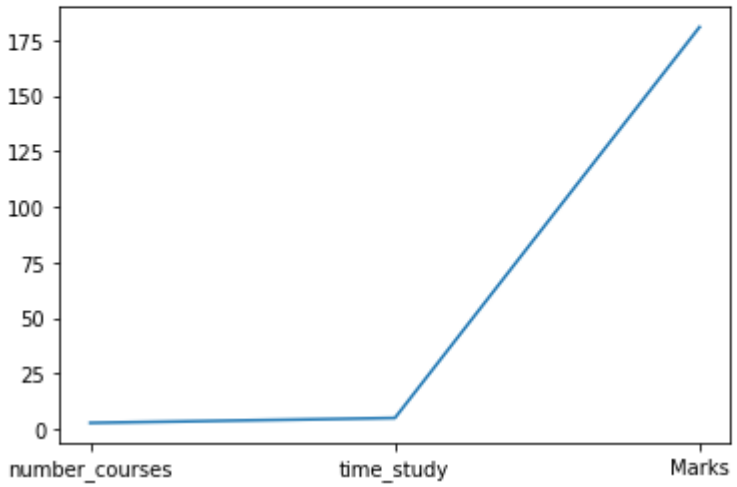
print(v)

plt.plot(v)

Output:

```
number_courses      2.859600
time_study          5.056281
Marks             180.929700
dtype: float64

[<matplotlib.lines.Line2D at 0x1ba835d9670>]
```



Standard Deviation:

sd=np.std(data)

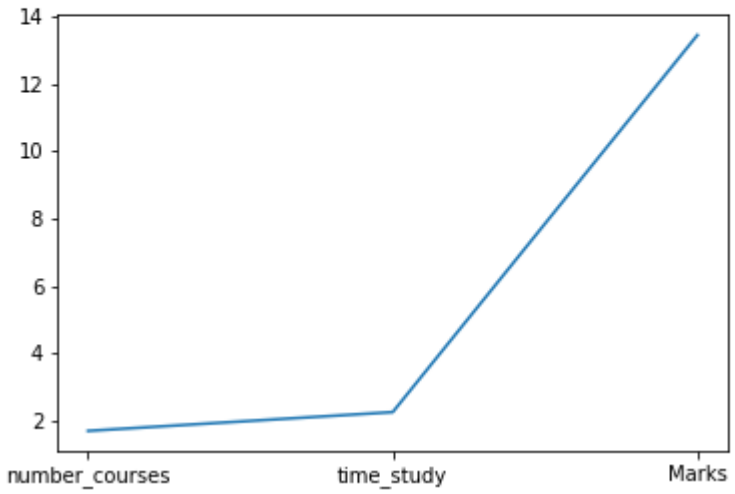
print(sd)

plt.plot(sd)

Output:

```
number_courses     1.691035
time_study         2.248617
Marks             13.451011
dtype: float64

[<matplotlib.lines.Line2D at 0x1ba83634b50>]
```



Ttest:

ttest,pvalue=stats.ttest_rel(data["number_courses"],data["time_study"])

if pvalue<0.05:

    print("reject the hypothesis")

else:

    print("accept the hypthesis")

print("ttest value:",ttest)

print("pvalue",pvalue)

Output:

reject the hypothesis
ttest value: 2.54623224109801
pvalue 0.014083405476284708

3.Aim: To perform Anova,Correlation and Chi Square.

Anova:

```
import  pandas as pd
from scipy import stats
from scipy.stats import chi2_contingency
data=pd.read_csv("C:/Users/Admin/Downloads/Student_Marks.csv").head(50)


f_oneway(data["number_courses"],data["time_study"])
```

output:

```
F_onewayResult(statistic=5.825455026664431,
pvalue=0.017653384997631253)
```

Correlation:

```
data.corr()
```

Output:

	number_courses	time_study	Marks
number_courses	1.000000	0.105615	0.346846
time_study	0.105615	1.000000	0.930313
Marks	0.346846	0.930313	1.000000

Chi Square:

```
stat,p,dof,expected=chi2_contingency(data)

alpha=0.05

print("p value is",str(p))

print(dof)

print(stat)

if p<=alpha:

    print("dependent (reject H0)")

else:
```

```
    print("independent (H0 holds True) ")
```

Output:

```
p value is 0.9106452240760612
98
79.76760237061065
independent (H0 holds True)
```