HARI PRASATH S

225229110

open csv file

In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv("mark.csv")
print(df)
```

```
  name  m1  m2  m3
0   Hp  80  50  45
1   Jp  70  67  80
2   Sp  75  70  45
3   Kp  70  80  60
```

size

In [3]:

```
size=df.size
print(size)
```

```
16
```

shape

In [4]:

```
shape=df.shape
print(shape)
```

```
(4, 4)
```

dim

In [5]:

```
dim=df.ndim
print(dim)
```

```
2
```

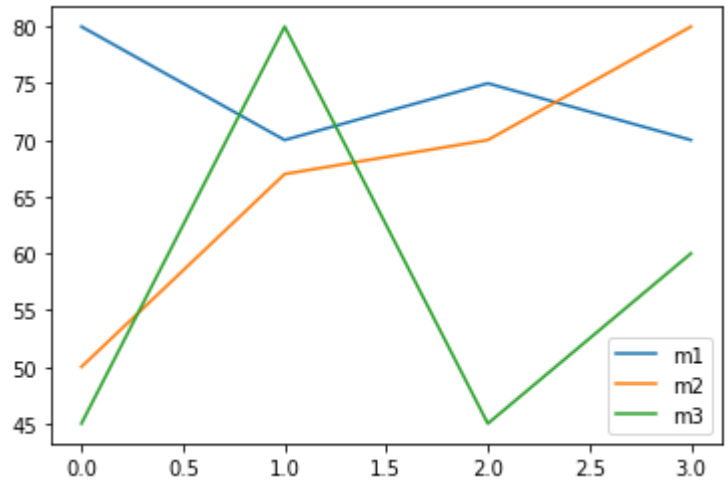
Plot

In [6]:

```
df.plot()
```

Out[6]:

<AxesSubplot:>



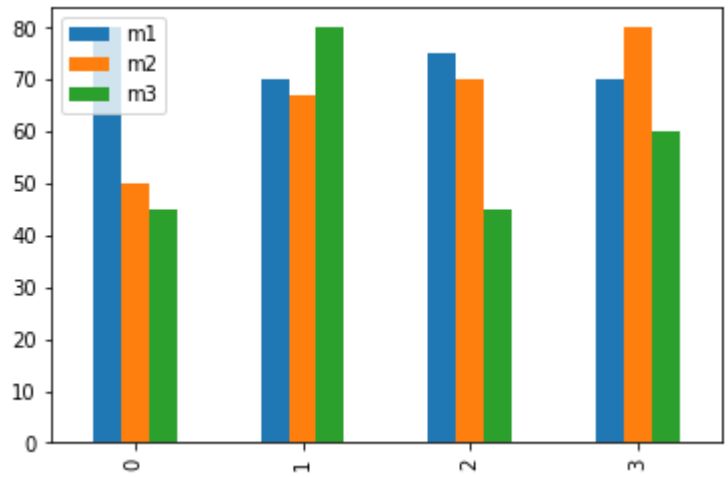
Barplot

In [7]:

```
df.plot(kind="bar")
```

Out[7]:

<AxesSubplot:>



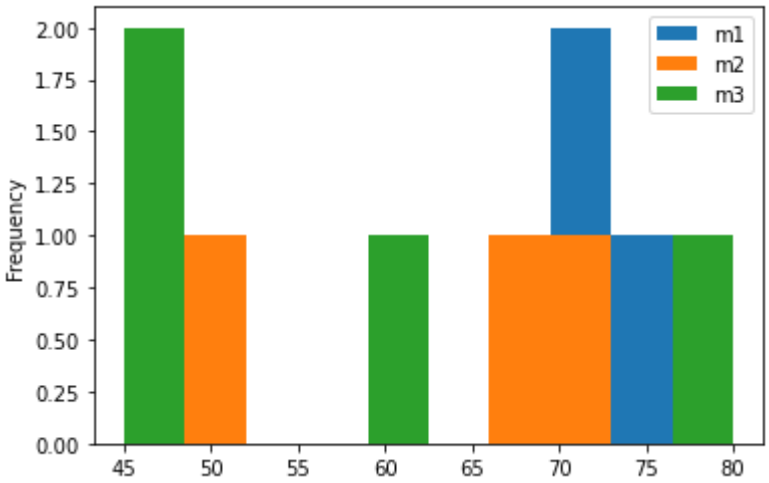
Histogram

In [8]:

```
df.plot.hist()
```

Out[8]:

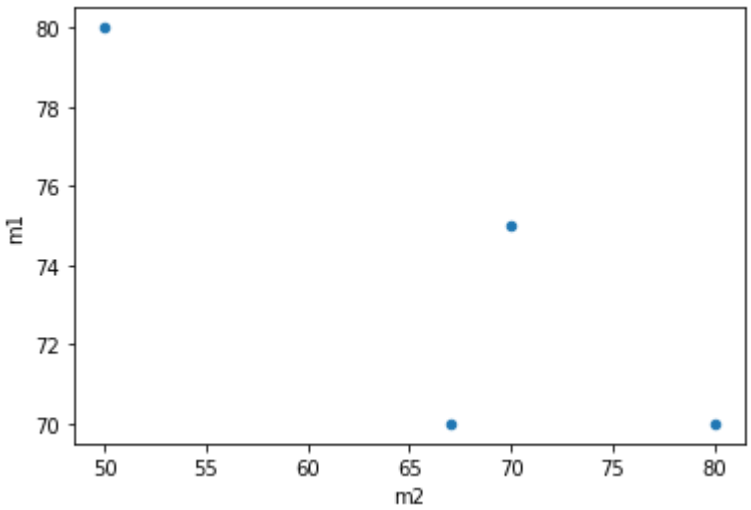
```
<AxesSubplot:ylabel='Frequency'>
```



scatter plot

In [9]:

```
df.plot.scatter(x='m2',y='m1');
```

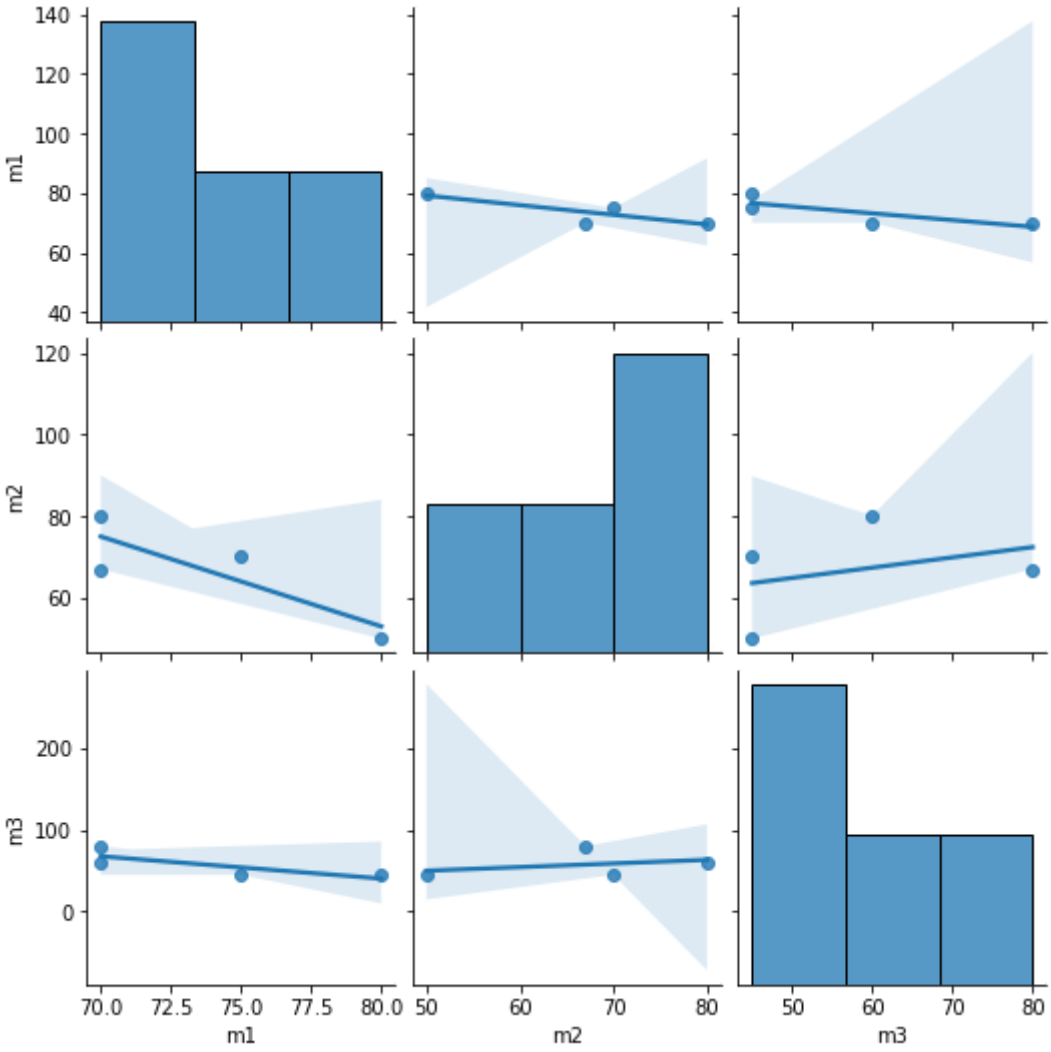


In [10]:

```
import seaborn as sns
```

In [11]:

```
h=sns.pairplot(df,kind='reg')
```



Pie chart

In [12]:

```
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(4,4))
Name=['Hp','Jp','Sp','Vp']
M1=[80,50,70,40]
plt.pie(M1,labels=Name)
plt.show()
```

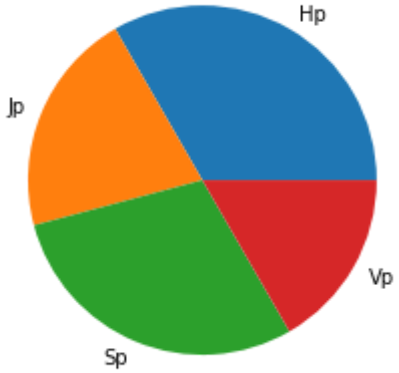


Image plot

In [17]:

```
import matplotlib.pyplot as plt
import matplotlib.image as img
pic = img.imread('dspic.png')
plt.imshow(pic)
```

Out[17]:

```
<matplotlib.image.AxesImage at 0x260184fe190>
```

In [18]:

```
print(pic.shape)
```

```
(153, 330, 4)
```

In [21]:

```
new = pic[:, :, 0]
```

Modified image display

In [22]:

```
plt.imshow(new)
```

Out[22]:

```
<matplotlib.image.AxesImage at 0x26018634040>
```

3D

In [23]:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(10, 10))
ax = plt.axes(projection='3d')
x = np.arange(0, 20, 0.1)
y = np.sin(x)
z = y*np.sin(x)
c = x + y
ax.scatter(x, y, z, c=c)
plt.axis('off')
plt.show()
```

In [3]:

```
import wave
audio=wave.open('hpaudio.wav','r')
print(audio)
```

```
<wave.Wave_read object at 0x00000161A1F64610>
```

In [4]:

```
pip install Ipython
```

```
Requirement already satisfied: Ipython in c:\users\lenovo\anaconda3\lib\site
-packages (8.2.0)
Requirement already satisfied: setuptools>=18.5 in c:\users\lenovo\anaconda3
\lib\site-packages (from Ipython) (61.2.0)
Requirement already satisfied: pickleshare in c:\users\lenovo\anaconda3\lib
\site-packages (from Ipython) (0.7.5)
Requirement already satisfied: stack-data in c:\users\lenovo\anaconda3\lib\s
ite-packages (from Ipython) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0
in c:\users\lenovo\anaconda3\lib\site-packages (from Ipython) (3.0.20)
Requirement already satisfied: backcall in c:\users\lenovo\anaconda3\lib\sit
e-packages (from Ipython) (0.2.0)
Requirement already satisfied: pygments>=2.4.0 in c:\users\lenovo\anaconda3
\lib\site-packages (from Ipython) (2.11.2)
Requirement already satisfied: traitlets>=5 in c:\users\lenovo\anaconda3\lib
\site-packages (from Ipython) (5.1.1)
Requirement already satisfied: matplotlib-inline in c:\users\lenovo\anaconda
3\lib\site-packages (from Ipython) (0.1.2)
Requirement already satisfied: decorator in c:\users\lenovo\anaconda3\lib\si
te-packages (from Ipython) (5.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\lenovo\anaconda3\lib\s
ite-packages (from Ipython) (0.18.1)
Requirement already satisfied: colorama in c:\users\lenovo\anaconda3\lib\sit
e-packages (from Ipython) (0.4.4)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\lenovo\anacon
da3\lib\site-packages (from jedi>=0.16->Ipython) (0.8.3)
Requirement already satisfied: wcwidth in c:\users\lenovo\anaconda3\lib\site
-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->Ipython) (0.2.
5)
Requirement already satisfied: pure-eval in c:\users\lenovo\anaconda3\lib\si
te-packages (from stack-data->Ipython) (0.2.2)
Requirement already satisfied: asttokens in c:\users\lenovo\anaconda3\lib\si
te-packages (from stack-data->Ipython) (2.0.5)
Requirement already satisfied: executing in c:\users\lenovo\anaconda3\lib\si
te-packages (from stack-data->Ipython) (0.8.3)
Requirement already satisfied: six in c:\users\lenovo\anaconda3\lib\site-pac
kages (from asttokens->stack-data->Ipython) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

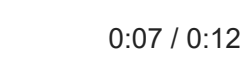
```
import IPython
```

opening mp3 file

In [4]:

```
IPython.display.Audio("audio.mp3")
```

Out[4]:



opening mp4 file

In [6]:

```
from IPython.display import Video
Video("sample.mp4")
```

Out[6]:

In [8]:

```
pip install docx2txt
```

```
Collecting docx2txt
  Downloading docx2txt-0.8.tar.gz (2.8 kB)
Building wheels for collected packages: docx2txt
  Building wheel for docx2txt (setup.py): started
  Building wheel for docx2txt (setup.py): finished with status 'done'
  Created wheel for docx2txt: filename=docx2txt-0.8-py3-none-any.whl size=39
84 sha256=27a1b0883f4eeca88cf6e4bb6f79831c5d62dee34c6bf319ee157be28510ee7a
  Stored in directory: c:\users\lenovo\appdata\local\pip\cache\wheels\40\75
\01\e6c444034338bde9c7947d3467807f889123465c2371e77418
Successfully built docx2txt
Installing collected packages: docx2txt
Successfully installed docx2txt-0.8
Note: you may need to restart the kernel to use updated packages.
```

word file operation

In [1]:

```
import docx2txt
hps=docx2txt.process("python.docx")
hps
```

Out[1]:

```
"Python is commonly used for\xa0developing websites and software, task autom
ation, data analysis, and data visualization. Since it's relatively easy to
learn, Python has been adopted by many non-programmers such as accountants a
nd scientists, for a variety of everyday tasks, like organizing finances"
```

Pdf

In [3]:

```
from IPython.display import IFrame,display
hp='python.pdf'
IFrame(hp,width=600,height=500)
```

Out[3]:

txtfile

In [4]:

```
f=open('textfile.txt','r')
f.read()
```

Out[4]:

```
'Hari is a good boy\nHe is persuing Msc Data science at Bishop Heber College
currently'
```

Tsv

In [5]:

```
import pandas as pd
data=pd.read_csv("mark.csv")
```

In [6]:

```
data.head
```

Out[6]:

```
<bound method NDFrame.head of   name  m1  m2  m3
0   Hp  80  50  45
1   Jp  70  67  80
2   Sp  75  70  45
3   Kp  70  80  60>
```

In [7]:

```
data.tail
```

Out[7]:

```
<bound method NDFrame.tail of   name  m1  m2  m3
0   Hp  80  50  45
1   Jp  70  67  80
2   Sp  75  70  45
3   Kp  70  80  60>
```

In [8]:

```
data.ndim
```

Out[8]:

```
2
```

In [9]:

```
data.size
```

Out[9]:

```
16
```

In [10]:

```
data.shape
```

Out[10]:

```
(4, 4)
```

In [ ]:

```

```