Python for Data Science LAB Session

->Working with Dataset using pandas

In [1]:
```
import numpy as np
import pandas as pd
```

Q-1) Load the data from elements.txt, titanic.csv, Values.xls into pandas datafram

In [2]:
```
txt = pd.read_fwf('elements.txt')   # fixed width file
txt
```

Out[2]:

	A_Wght\tSymbol\tName
0	1.008\tH\tHydrogen
1	4.003\tHe\tHelium
2	6.941\tLi\tLithium
3	9.012\tBe\tBeryllium
4	10.811\tB\tBoron
5	12.011\tC\tCarbon
6	14.007\tN\tNitrogen
7	15.999\tO\tOxygen
8	18.998\tF\tFluorine
9	20.180\tNe\tNeon

In [3]:

```
txt1 = pd.read_table('elements.txt')  # table format
txt1
```

Out[3]:

	A_Wght	Symbol	Name
0	1.008	H	Hydrogen
1	4.003	He	Helium
2	6.941	Li	Lithium
3	9.012	Be	Beryllium
4	10.811	B	Boron
5	12.011	C	Carbon
6	14.007	N	Nitrogen
7	15.999	O	Oxygen
8	18.998	F	Fluorine
9	20.180	Ne	Neon

In [4]:

```
df = pd.read_csv('titanic.csv')  #csv file for comma seprated value
df
```

Out[4]:

	Unnamed: 0	pclass	survived	sex	age	sibsp	parch
0	1	1st	survived	female	29.0000	0	0
1	2	1st	survived	male	0.9167	1	2
2	3	1st	died	female	2.0000	1	2
3	4	1st	died	male	30.0000	1	2
4	5	1st	died	female	25.0000	1	2
...	...	...	...	...	...	...	...
1304	1305	3rd	died	female	14.5000	1	0
1305	1306	3rd	died	female	9999.0000	1	0
1306	1307	3rd	died	male	26.5000	0	0
1307	1308	3rd	died	male	27.0000	0	0
1308	1309	3rd	died	male	29.0000	0	0

1309 rows × 7 columns

In [5]:

```
data = pd.read_excel('Values.xls')
data
```

Out[5]:

	Angle (Degrees)	Sine	Cosine	Tangent
0	138.550574	0.661959	-0.749540	-0.883153
1	305.535745	-0.813753	0.581211	-1.400100
2	280.518695	-0.983195	0.182556	-5.385709
3	216.363795	-0.592910	-0.805269	0.736289
4	36.389247	0.593268	0.805005	0.736974
...	...	...	...	...
67	324.199562	-0.584964	0.811059	-0.721234
68	187.948172	-0.138277	-0.990394	0.139619
69	270.678249	-0.999930	0.011837	-84.472139
70	270.779159	-0.999908	0.013598	-73.530885
71	200.213513	-0.345520	-0.938412	0.368196

72 rows × 4 columns

Q-2) Load the image Colorblk.jpg, check the size of original image, convert it to the 1D data and check the size now. Resize the image to 50x50 grey-scale image, convert to 1D and check the size of it.

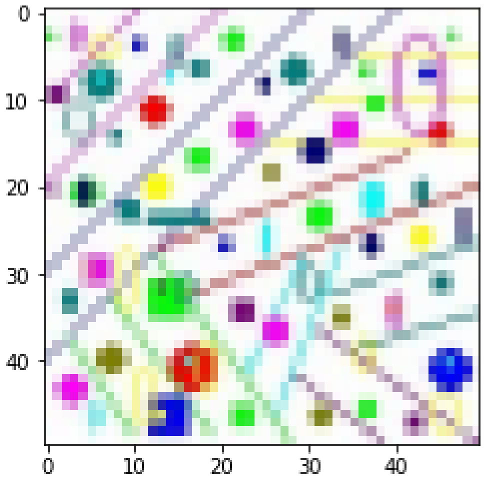
In [6]:

```
from PIL import Image

img = Image.open('colorblk.jpg')


print(img.size) # find the size of image
```

```
(100, 100)
```

```
In [7]: import matplotlib.image as img
        import matplotlib.pyplot as plt
        from skimage.io import imread
        from skimage.transform import resize

        # Read Images
        img1 = img.imread('colorblk.jpg')

        plt.imshow(img1)
        print(img1.shape)  # size of image



        I1 = img1.flatten()
        print(I1.shape)  # shape of imahe


        # resize imag in  50*50
        resize1 = resize(img1, (50,50))  # resize image
        plt.imshow(resize1)
        print(resize1.shape)
```

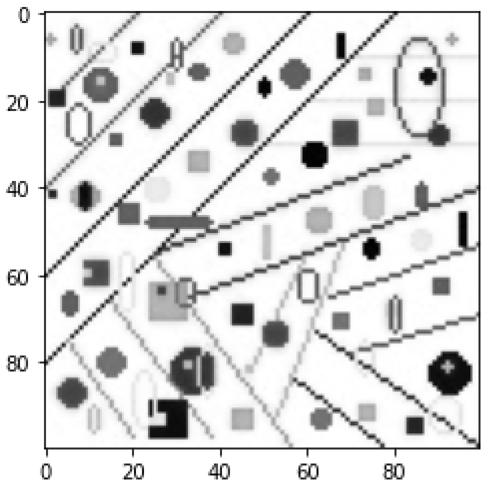
```
(100, 100, 3)
(30000,)
(50, 50, 3)
```



In [8]:
```
from skimage.color import rgb2gray

gray  = rgb2gray(img1)
plt.imshow(gray, cmap = 'gray')   # give color from cmap =>
#plt.imshow(gray, cmap='Greys_r')

print(gray.shape)
gray = rgb2gray(gray)
gray = gray.flatten()  # reshape in 1d array
print(gray.shape)
```

```
(100, 100)
(10000,)
```

```
c:\users\arjun vankani\appdata\local\programs\python\python37\lib\site-packag
es\ipykernel_launcher.py:8: FutureWarning: The behavior of rgb2gray will chan
ge in scikit-image 0.19. Currently, rgb2gray allows 2D grayscale image to be
passed as inputs and leaves them unmodified as outputs. Starting from version
0.19, 2D arrays will be treated as 1D images with 3 channels.
```



3. From titanic.csv, find the number of male passengers who were travelling in 2nd class that survived.

```
In [9]: df
```

Out[9]:

	Unnamed: 0	pclass	survived	sex	age	sibsp	parch
0	1	1st	survived	female	29.0000	0	0
1	2	1st	survived	male	0.9167	1	2
2	3	1st	died	female	2.0000	1	2
3	4	1st	died	male	30.0000	1	2
4	5	1st	died	female	25.0000	1	2
...	...	...	...	...	...	...	...
1304	1305	3rd	died	female	14.5000	1	0
1305	1306	3rd	died	female	9999.0000	1	0
1306	1307	3rd	died	male	26.5000	0	0
1307	1308	3rd	died	male	27.0000	0	0
1308	1309	3rd	died	male	29.0000	0	0

1309 rows × 7 columns

In [10]:

```
# find the 2nd class passenger
class_2 = df["pclass"]=="2nd"
df=df.loc[class_2]       # c1 => class_2 a => df  c2 = c_2 c=>df1   c3 => c_2nd
b = df_final


# which are survided male
c_2=df["survived"]=="survived"
df1 = df.loc[c_2]
c_2nd = df1["sex"]=="male"
df_final=df1.loc[c_2nd]
df_final
```

Out[10]:

	Unnamed: 0	pclass	survived	sex	age	sibsp	parch
336	337	2nd	survived	male	32.0000	1	0
339	340	2nd	survived	male	1.0000	2	1
343	344	2nd	survived	male	34.0000	0	0
359	360	2nd	survived	male	0.8333	0	2
360	361	2nd	survived	male	26.0000	1	1
376	377	2nd	survived	male	24.0000	0	0
385	386	2nd	survived	male	8.0000	1	1
398	399	2nd	survived	male	8.0000	0	2
427	428	2nd	survived	male	0.6667	1	1
432	433	2nd	survived	male	62.0000	0	0
454	455	2nd	survived	male	42.0000	0	0
492	493	2nd	survived	male	1.0000	0	2
503	504	2nd	survived	male	19.0000	0	0
514	515	2nd	survived	male	2.0000	1	1
515	516	2nd	survived	male	3.0000	1	1
520	521	2nd	survived	male	20.0000	0	0
523	524	2nd	survived	male	22.0000	0	0
524	525	2nd	survived	male	9999.0000	0	0
526	527	2nd	survived	male	29.0000	0	0
538	539	2nd	survived	male	30.0000	0	0
548	549	2nd	survived	male	0.8333	1	1
549	550	2nd	survived	male	3.0000	1	1
587	588	2nd	survived	male	2.0000	1	1
596	597	2nd	survived	male	31.0000	0	0
597	598	2nd	survived	male	9999.0000	0	0

Q-4) From Values.xls, create the dataframe that contain the cosine values of the angles between 150 degree to 200 degree.

In [11]:

```
data
```

Out[11]:

	Angle (Degrees)	Sine	Cosine	Tangent
0	138.550574	0.661959	-0.749540	-0.883153
1	305.535745	-0.813753	0.581211	-1.400100
2	280.518695	-0.983195	0.182556	-5.385709
3	216.363795	-0.592910	-0.805269	0.736289
4	36.389247	0.593268	0.805005	0.736974
...	...	...	...	...
67	324.199562	-0.584964	0.811059	-0.721234
68	187.948172	-0.138277	-0.990394	0.139619
69	270.678249	-0.999930	0.011837	-84.472139
70	270.779159	-0.999908	0.013598	-73.530885
71	200.213513	-0.345520	-0.938412	0.368196

72 rows × 4 columns

In [12]:

```
# find cosine value in table which > 150
col1=data["Angle (Degrees)"]>150
data1=data.loc[col1]
col2=data1["Angle (Degrees)"]<200
ans = data1.loc[col2]
ans.loc[:,("Cosine")]
```

Out[12]:

```
8    -0.999998
14   -0.994096
24   -0.955297
30   -0.981569
33   -0.998929
37   -0.999200
38   -0.954926
41   -0.984736
47   -0.965127
52   -0.951402
54   -0.997873
62   -0.978785
64   -0.954533
68   -0.990394
Name: Cosine, dtype: float64
```

Q-5) Create a dataframe from last 5 elements from elements.txt and find their average atomic weight

In [13]:

```
print(txt1)  #data
print(txt1.tail(5))    #last 5 element
last5 = txt1.tail(5)
np.average(last5["A_Wght"]) # average
```

```
   A_Wght Symbol       Name
0   1.008      H   Hydrogen
1   4.003     He     Helium
2   6.941     Li    Lithium
3   9.012     Be  Beryllium
4  10.811      B      Boron
5  12.011      C     Carbon
6  14.007      N   Nitrogen
7  15.999      O     Oxygen
8  18.998      F   Fluorine
9  20.180     Ne       Neon
   A_Wght Symbol      Name
5  12.011      C    Carbon
6  14.007      N  Nitrogen
7  15.999      O    Oxygen
8  18.998      F  Fluorine
9  20.180     Ne      Neon
```

Out[13]:

```
16.238999999999997
```

In [ ]: