NAME: ANNAPOORNIMA.S

ROLL NO: 225229101

Red Wine Quality Data Analysis using NumPy Part-I ¶

import modules for numpy

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

In [4]:
```
wines = np.genfromtxt("winequality-red.csv",delimiter=";",skip_header=1)
```

What is its shape

In [46]:
```
wines.shape
```

Out[46]: (1599, 12)

How many wine data rows here?

In [11]:
```
wines.shape[0]
```

Out[11]: 1599

How many wine data columns here?

In [13]:
```
wines.shape[1]
```

Out[13]: 12

How many dimensions?

In [14]:
```
wines.ndim
```

Out[14]: 2

What is the type of wines?

In [15]:
```
type(wines)
```

Out[15]:
```
numpy.ndarray
```

What is the data type of wines data?

In [16]:
```
wines.dtype
```

Out[16]:
```
dtype('float64')
```

Show top 5 rows

In [17]:
```
#wines[:5,;]
```

What is the value at 3rd row, 4th column of wine data?

In [18]:
```
wines[2,3]
```

Out[18]:
```
2.3
```

Select first 3 items in 4th column

In [19]:
```
wines[:3,3]
```

Out[19]:
```
array([1.9, 2.6, 2.3])
```

Show 1st column

In [20]:
```
wines[:,0]
```

Out[20]:
```
array([7.4, 7.8, 7.8, ..., 6.3, 5.9, 6. ])
```

Show 2nd row

In [21]:
```
wines[1,:]
```

Out[21]:
```
array([ 7.8   ,  0.88  ,  0.    ,  2.6   ,  0.098 , 25.    , 67.    ,
        0.9968,  3.2   ,  0.68  ,  9.8   ,  5.    ])
```

Select items from rows 1 to 3 and 5th column

In [22]:
```
wines[1:4,4]
```

Out[22]:
```
array([0.098, 0.092, 0.075])
```

Select entire array

Select entire array

In [23]:
```
wines[:,:]
```

Out[23]:
```
array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  ,  5.   ],
       [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  ,  5.   ],
       [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  ,  5.   ],
       ...,
       [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   ,  6.   ],
       [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  ,  5.   ],
       [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   ,  6.   ]])
```

Change 1st value in wines to 100

In [27]:
```
wines[0,0]
```

Out[27]:
```
7.4
```

In [28]:
```
wines[0,0]=100
wines[0,0]
```

Out[28]:
```
100.0
```

#change it back to 7.4 and print

In [29]:
```
wines[0,0]=7.4
```

1-Dimensional Numpy Arrays

Select 4th row all column values

In [30]:
```
third_wine=wines[3,:]
```

display its value

In [31]:
```
third_wine
```

Out[31]:
```
array([11.2  ,  0.28 ,  0.56 ,  1.9  ,  0.075, 17.   , 60.   ,  0.998,
        3.16 ,  0.58 ,  9.8  ,  6.   ])
```

show 2nd value

In [32]:
```
third_wine[1]
```

Out[32]:
```
0.28
```

convert wine data to integer values and show it

In [33]:
```
wines.astype(int)
```

Out[33]:
```
array([[ 7,  0,  0, ...,  0,  9,  5],
       [ 7,  0,  0, ...,  0,  9,  5],
       [ 7,  0,  0, ...,  0,  9,  5],
       ...,
       [ 6,  0,  0, ...,  0, 11,  6],
       [ 5,  0,  0, ...,  0, 10,  5],
       [ 6,  0,  0, ...,  0, 11,  6]])
```

Vectorization Operations

Increase wine quality score (output variable) by 10

In [34]:
```
wines[:,11]
```

Out[34]:
```
array([5., 5., 5., ..., 6., 5., 6.])
```

Increase by 10

In [35]:
```
wines[:,11]+=10
```

Display update score

In [36]:
```
wines[:,10]
```

Out[36]:
```
array([ 9.4,  9.8,  9.8, ..., 11. , 10.2, 11. ])
```

Multiply alcohol of all wine data by 3 times

In [37]:
```
wines[:,10]*=3
```

Show updated alcohol column

In [38]:
```
wines[:,10]
```

Out[38]:
```
array([28.2, 29.4, 29.4, ..., 33. , 30.6, 33. ])
```

Add quality column by itselt

In [39]:
```
wines[:,11]+wines[:,11]
```

Out[39]:
```
array([30., 30., 30., ..., 32., 30., 32.])
```

Multiply alcohol and wine quality columns. It will perform element wise multiplication

In [42]:
```
wines[:,10] * wines[:,11]
```

Out[42]:
```
array([225., 225., 225., ..., 256., 225., 256.])
```

Broadcasting

Add every row of wines data with a random array of values

In [43]:
```
rand_array = np.random.rand(12)
```

Show rand_array

In [44]:
```
rand_array
```

Out[44]:
```
array([0.50287562, 0.95298949, 0.31690841, 0.9159978 , 0.5625847 ,
       0.52995611, 0.5964002 , 0.49162233, 0.65293319, 0.74715201,
       0.26610582, 0.92630738])
```

Add wines and rand_array

In [45]:
```
wines + rand_array
```

Out[45]:
```
array([[ 7.90287562,  1.65298949,  0.31690841, ...,  1.30715201,
         15.26610582, 15.92630738],
       [ 8.30287562,  1.83298949,  0.31690841, ...,  1.42715201,
         15.26610582, 15.92630738],
       [ 8.30287562,  1.71298949,  0.35690841, ...,  1.39715201,
         15.26610582, 15.92630738],
       ...,
       [ 6.80287562,  1.46298949,  0.44690841, ...,  1.49715201,
         16.26610582, 16.92630738],
       [ 6.40287562,  1.59798949,  0.43690841, ...,  1.45715201,
         15.26610582, 15.92630738],
       [ 6.50287562,  1.26298949,  0.78690841, ...,  1.40715201,
         16.26610582, 16.92630738]])
```