ROLL NO : 205229133

NAME : VIVIYAN RICHARDS W

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

In [2]:
```
wines=np.genfromtxt("winequality-red.csv",delimiter=";",skip_header=1)
```

what is its shape

In [3]:
```
wines.shape
```

Out[3]: (1599, 12)

How many wine data rows here?

In [4]:
```
wines.shape[0]
```

Out[4]: 1599

How many wine data columns here?

In [5]:
```
wines.shape[1]
```

Out[5]: 12

How many dimensions?

In [6]:
```
 wines.ndim
```

Out[6]: 2

What is the type of data?

In [7]:
```
type(wines)
```

Out[7]: numpy.ndarray

What is the data type of wines data?

In [8]:
```
wines.dtype
```

Out[8]:
```
dtype('float64')
```

Show top 5 rows

In [9]:
```
wines[:5, :]
```

Out[9]:
```
array([[7.400e+00, 7.000e-01, 0.000e+00, 1.900e+00, 7.600e-02, 1.100e+01,
        3.400e+01, 9.978e-01, 3.510e+00, 5.600e-01, 9.400e+00, 5.000e+00],
       [7.800e+00, 8.800e-01, 0.000e+00, 2.600e+00, 9.800e-02, 2.500e+01,
        6.700e+01, 9.968e-01, 3.200e+00, 6.800e-01, 9.800e+00, 5.000e+00],
       [7.800e+00, 7.600e-01, 4.000e-02, 2.300e+00, 9.200e-02, 1.500e+01,
        5.400e+01, 9.970e-01, 3.260e+00, 6.500e-01, 9.800e+00, 5.000e+00],
       [1.120e+01, 2.800e-01, 5.600e-01, 1.900e+00, 7.500e-02, 1.700e+01,
        6.000e+01, 9.980e-01, 3.160e+00, 5.800e-01, 9.800e+00, 6.000e+00],
       [7.400e+00, 7.000e-01, 0.000e+00, 1.900e+00, 7.600e-02, 1.100e+01,
        3.400e+01, 9.978e-01, 3.510e+00, 5.600e-01, 9.400e+00, 5.000e+00]])
```

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

In [10]:
```
wines[2,3]
```

Out[10]:
```
2.3
```

Select first 3 items in 4th column

In [11]:
```
wines[:3, 3]
```

Out[11]:
```
array([1.9, 2.6, 2.3])
```

Show 1st column

In [12]:
```
wines[:, 0]
```

Out[12]:
```
array([7.4, 7.8, 7.8, ..., 6.3, 5.9, 6. ])
```

Show 2nd row

In [13]:
```
wines[1, :]
```

Out[13]:
```
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 [14]:
```
wines[1:4, 4]
```

Out[14]:
```
array([0.098, 0.092, 0.075])
```

Select entire array

In [15]:
```
wines[:,:]
```

Out[15]:
```
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 [16]:
```
 wines[0,0]
```

Out[16]:
```
7.4
```

In [17]:
```
wines[0,0] = 100
```

In [18]:
```
wines[0,0]
```

Out[18]:
```
100.0
```

In [19]:
```
# change it back
wines[0,0] = 7.4
```

0.1.1 1-Dimensional Numpy Arrays

Select 4th row all column values

In [20]:
```
third_wine = wines[3, :]
```

In [21]:
```
third_wine
```

Out[21]:
```
array([11.2  ,  0.28 ,  0.56 ,  1.9  ,  0.075, 17.   , 60.   ,  0.998,
        3.16 ,  0.58 ,  9.8  ,  6.   ])
```

In [22]:
```
third_wine[1]
```

Out[22]:
```
0.28
```

Convert one datatype to another

In [23]:
```
#convert to int
wines.astype(int)
```

Out[23]:
```
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]])
```

0.1.2 Vectorization Operations

Increase wine quality score (output variable) by 10

In [24]:
```
# check values first
wines[:, 11]
```

Out[24]:
```
array([5., 5., 5., ..., 6., 5., 6.])
```

In [25]:
```
 wines[:, 11] += 10
```

In [26]:
```
wines[:, 11]
```

Out[26]:
```
array([15., 15., 15., ..., 16., 15., 16.])
```

Multiply alcohol of all wine data by 3 times

In [27]:
```
wines[:, 10] *= 3
```

In [28]:
```
wines[:, 10]
```

Out[28]:
```
array([28.2, 29.4, 29.4, ..., 33. , 30.6, 33. ])
```

Add quality column by itselt

In [29]:
```
# It will produce a new array
wines[:, 11] + wines[:, 11]
```

Out[29]:
```
array([30., 30., 30., ..., 32., 30., 32.])
```

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

In [30]:
```
 wines[:,10] * wines[:,11]
```

Out[30]:
```
array([423., 441., 441., ..., 528., 459., 528.])
```

0.1.3 Broadcasting

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

In [31]:
```
rand_array = np.random.rand(12)
```

In [32]:
```
rand_array
```

Out[32]:
```
array([0.79517035, 0.32779929, 0.50497714, 0.54776994, 0.68385131,
       0.16661872, 0.32094485, 0.762968  , 0.25914871, 0.07375356,
       0.9678116 , 0.7019941 ])
```

In [33]:
```
wines + rand_array
```

Out[33]:
```
array([[ 8.19517035,  1.02779929,  0.50497714, ...,  0.63375356,
         29.1678116 , 15.7019941 ],
       [ 8.59517035,  1.20779929,  0.50497714, ...,  0.75375356,
         30.3678116 , 15.7019941 ],
       [ 8.59517035,  1.08779929,  0.54497714, ...,  0.72375356,
         30.3678116 , 15.7019941 ],
       ...,
       [ 7.09517035,  0.83779929,  0.63497714, ...,  0.82375356,
         33.9678116 , 16.7019941 ],
       [ 6.69517035,  0.97279929,  0.62497714, ...,  0.78375356,
         31.5678116 , 15.7019941 ],
       [ 6.79517035,  0.63779929,  0.97497714, ...,  0.73375356,
         33.9678116 , 16.7019941 ]])
```