

Name:P.Asha Belcilda

Roll No:225229104

Lab-1 Red Wine Quality Data Analytics using NumPy Part-1¶

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

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

What is its size?

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

```
Out[3]: 19188
```

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 wines?

```
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
```

change it back to 7.4 and print

```
In [19]: wines[0,0] = 7.4
         wines[0,0]
```

```
Out[19]: 7.4
```

1-Dimensional Numpy Array

Select 4th row all column values

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

Display its value

```
In [21]: wines[3,:]
```

```
Out[21]: 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 [22]: fourth_row[1]
```

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

Convert wine data to integer values and show it

```
In [23]: 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]])
```

Vectorization Operations

Increase wine quality score (output variable) by 10

```
In [24]: wines[:, 11]
```

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

Increases by 10

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

Display update score

```
In [27]: wines[:,11]
```

```
Out[27]: array([15., 15., 15., ..., 16., 15., 16.])
```

Multiply alcohol of all wine data by 3 times

```
In [28]: wines[:, 10] *= 3
```

Show updated alcohol column

```
In [29]: wines[:, 10]
```

```
Out[29]: array([28.2, 29.4, 29.4, ..., 33. , 30.6, 33. ])
```

Add quality column by itsel

```
In [30]: wines[:, 11] + wines[:, 11]
```

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

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

```
In [31]: wines[:,10] * wines[:,11]
```

```
Out[31]: array([423., 441., 441., ..., 528., 459., 528.])
```

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

```
In [32]: rand_array = np.random.rand(12)
```

show rand_array

```
In [33]: rand_array
```

```
Out[33]: array([0.71301706, 0.95074836, 0.46888046, 0.11508173, 0.52816029,  
               0.27284099, 0.91548833, 0.96699258, 0.01930503, 0.16459068,  
               0.5376563 , 0.22085147])
```

add wines and rand_array

```
In [35]: wines + rand_array
```

```
Out[35]: array([[ 8.11301706,  1.65074836,  0.46888046, ...,  0.72459068,  
                 28.7376563 , 15.22085147],  
               [ 8.51301706,  1.83074836,  0.46888046, ...,  0.84459068,  
                 29.9376563 , 15.22085147],  
               [ 8.51301706,  1.71074836,  0.50888046, ...,  0.81459068,  
                 29.9376563 , 15.22085147],  
               ...,  
               [ 7.01301706,  1.46074836,  0.59888046, ...,  0.91459068,  
                 33.5376563 , 16.22085147],  
               [ 6.61301706,  1.59574836,  0.58888046, ...,  0.87459068,  
                 31.1376563 , 15.22085147],  
               [ 6.71301706,  1.26074836,  0.93888046, ...,  0.82459068,  
                 33.5376563 , 16.22085147]])
```