HARI PRASATH S

225229110

RED WINE QUALITY DATA ANALYSIS USING NUMPY PART-2

In [ ]:

```
#Importing Necessary Modules
```

In [41]:

```
import numpy as np
```

In [42]:

```
wines = np.genfromtxt("winequality-red.csv", delimiter=";", skip_header=1)
```

Sum Of All Residual Sugar Values

In [3]:

```
y=wines[:,3]
sum(y)
```

Out[3]:

```
4059.550000000003
```

Sum Of Every Feature Value

In [4]:

```
sum(wines)
```

Out[4]:

```
array([13303.1    ,   843.985  ,   433.29   ,  4059.55   ,   139.859  ,
       25384.     , 74302.     ,  1593.79794,  5294.47   ,  1052.38   ,
       16666.35   ,  9012.     ])
```

Sum Of Every Row

In [5]:

```
z=wines[:,:].sum(axis=1)
z
```

Out[5]:

```
array([ 74.5438 , 123.0548 ,  99.699  , ..., 100.48174, 105.21547,
        92.49249])
```

Size

In [6]:

```
wines.shape[0]
```

Out[6]:

```
1599
```

Maximum Residual Sugar value in red wines data

In [7]:

```
p=wines[:,3]
p=y.astype('int32')
p
```

Out[7]:

```
array([1, 2, 2, ..., 2, 2, 3])
```

Maximum Residual Sugar value

In [8]:

```
max(p)
```

Out[8]:

```
15
```

Minimum Residual Sugar value

In [9]:

```
min(p)
```

Out[9]:

```
0
```

Average Residual Sugar value in red wines data

In [10]:

```
np.mean(y)
```

Out[10]:

```
2.53880550343965
```

25 percentile residual sugar value

In [11]:

```
np.percentile(y,25)
```

Out[11]:

```
1.9
```

75 percentile residual sugar value

In [12]:

```
np.percentile(y,75)
```

Out[12]:

```
2.6
```

Average of each feature value

In [13]:

```
t=wines[:,:]
t
t.mean(axis=0)
```

Out[13]:

```
array([ 8.31963727,  0.52782051,  0.27097561,  2.5388055 ,  0.08746654,
       15.87492183, 46.46779237,  0.99674668,  3.3111132 ,  0.65814884,
       10.42298311,  5.63602251])
```

Numpy Array Comparisons

Wines with quality >5

In [14]:

```
wines[:,11]>5
```

Out[14]:

```
array([False, False, False, ...,  True, False,  True])
```

Wines with quality >7

In [15]:

```
k=wines[:,11]>7
k
```

Out[15]:

```
array([False, False, False, ..., False, False, False])
```

True for the condition quality >7

In [16]:

```
True in k
```

Out[16]:

```
True
```

First 3 rows whetre wine quality >7,call it high_quality

In [17]:

```
high_quality=wines[:,11]>7
high_quality
```

Out[17]:

```
array([False, False, False, ..., False, False, False])
```

Only top 3 rows and columns of high_quality wines data

In [18]:

```
wines[high_quality][0:3]
```

Out[18]:

```
array([[7.900e+00, 3.500e-01, 4.600e-01, 3.600e+00, 7.800e-02, 1.500e+01,
        3.700e+01, 9.973e-01, 3.350e+00, 8.600e-01, 1.280e+01, 8.000e+00],
       [1.030e+01, 3.200e-01, 4.500e-01, 6.400e+00, 7.300e-02, 5.000e+00,
        1.300e+01, 9.976e-01, 3.230e+00, 8.200e-01, 1.260e+01, 8.000e+00],
       [5.600e+00, 8.500e-01, 5.000e-02, 1.400e+00, 4.500e-02, 1.200e+01,
        8.800e+01, 9.924e-01, 3.560e+00, 8.200e-01, 1.290e+01, 8.000e+00]])
```

Wines with alcohol >10 and high wine quality >7

In [19]:

```
alcohol=wines[:,10]>10
alcohol
```

Out[19]:

```
array([False, False, False, ...,  True,  True,  True])
```

In [20]:

```
h=alcohol & k
h
```

Out[20]:

```
array([False, False, False, ..., False, False, False])
```

Alcohol and wine quality columns

In [21]:

```
wines[h,10:]
```

Out[21]:

```
array([[12.8,  8. ],
       [12.6,  8. ],
       [12.9,  8. ],
       [13.4,  8. ],
       [11.7,  8. ],
       [11. ,  8. ],
       [11. ,  8. ],
       [14. ,  8. ],
       [12.7,  8. ],
       [12.5,  8. ],
       [11.8,  8. ],
       [13.1,  8. ],
       [11.7,  8. ],
       [14. ,  8. ],
       [11.3,  8. ],
       [11.4,  8. ]])
```

Combining NumPy Arrays

Combine red wine and white wine data

Open White wine dataset

In [22]:

```
white_wines=np.genfromtxt("winequality-white.csv",delimiter=";",skip_header=1)
```

Size of White_wines

In [23]:

```
white_wines.shape
```

Out[23]:

```
(4898, 12)
```

Combine wines and white_wines data frames using vstack and call it all_wines

In [24]:

```
all_wines=np.vstack((wines,white_wines))
```

In [25]:

```
all_wines.shape
```

Out[25]:

```
(6497, 12)
```

Combine wines and white_wines data frames using concatenate method

In [26]:

```
all_wines1=np.concatenate((wines,white_wines),axis=0)
```

In [27]:

```
all_wines1.shape
```

Out[27]:

```
(6497, 12)
```

Transpose of wines and its size

In [28]:

```
tran=wines.T
tran.shape
```

Out[28]:

```
(12, 1599)
```

Convert wines data into 1D array

In [29]:

```
wines.ravel()
```

Out[29]:

```
array([ 7.4 ,  0.7 ,  0.  , ...,  0.66, 11.  ,  6.  ])
```

Size

In [30]:

```
wines.ravel().shape
```

Out[30]:

```
(19188,)
```

Reshape second row of wines into a 2-dim array with 2rows and 6 columns

In [31]:

```
wines[1].reshape((2,6))
```

Out[31]:

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

Sort alcohol column ascending order

In [34]:

```
sorted_alcohol=np.sort(wines[:,-2])
```

In [35]:

```
sorted_alcohol
```

Out[35]:

```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```

Make sorting to take place in-place

In [36]:

```
wines[:,-2].sort()
```

In [37]:

```
wines[:,-2]
```

Out[37]:

```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```

Sort alcohol column Descending order

In [38]:

```
sorted_alcohol_desc=np.sort(wines[:,10])[::-1]
```

In [39]:

```
sorted_alcohol_desc
```

Out[39]:

```
array([14.9, 14. , 14. , ...,  8.5,  8.4,  8.4])
```

will original data be modified?

In [40]:

```
wines[:,-2]
```

Out[40]:

```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```

In [ ]:

```

```