NAME : RETHINAGIRI G

ROLL NO : 225229130

COURSE TITLE : Data and Visual Analytics Lab

LAB : 02 Red Wine Quality Data Analysis using Numpy Part-II

Import necessary modules

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

In [2]:
```
wine=num.genfromtxt("C:\\Users\\user\\Downloads\\winequality-red.csv",delimiter=";",skip_header=1)
```

NumPy Aggregation Methods

Find sum of all residual sugar values

In [3]:
```
wine[:,3].sum()
```

Out[3]:
```
4059.55
```

Find sums of every feature value. There are 12 features altogether

In [4]:
```
wine.sum(axis=0)
```

Out[4]:
```
array([13303.1    ,   843.985  ,   433.29   ,  4059.55   ,   139.859  ,
       25384.     , 74302.     ,  1593.79794,  5294.47   ,  1052.38   ,
       16666.35   ,  9012.     ])
```

Find sum of every row

In [5]:
```
wine.sum(axis=1)
```

Out[5]:
```
array([ 74.5438 , 123.0548 ,  99.699  , ..., 100.48174, 105.21547,
        92.49249])
```

What is its size?

In [6]:
```
wine.shape[0]
```

Out[6]:
```
1599
```

What is the maximum residual sugar value in red wines data?

In [7]:
```
wine[:,3].astype("int32")
```

Out[7]:
```
array([1, 2, 2, ..., 2, 2, 3])
```

find its maximum residual sugar value

In [8]:
```
int(wine[:,3].max())
```

Out[8]:
```
15
```

What is the minimum residual sugar value in red wines data?

In [9]:
```
int(wine[:,3].min())
```

Out[9]:
```
0
```

What is the average residual sugar value in red wines data?

In [10]:
```
sum(wine[:,3])/len(wine[:,3])
```

Out[10]:
```
2.5388055034396517
```

What is 25 percentile residual sugar value?

In [11]:
```
num.percentile(wine[:,3],25)
```

Out[11]:
```
1.9
```

What is 75 percentile residual sugar value?

In [12]:
```
num.percentile(wine[:,3],75)
```

Out[12]:
```
2.6
```

Find the average of each feature value

In [13]:
```
sum(wine)/len(wine)
```

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

Show all wines with quality > 5

In [14]:
```
wine[:,11]>5
```

Out[14]:
```
array([False, False, False, ...,  True, False,  True])
```

Show all wines with quality > 7

In [15]:
```
wine[:,11]>6
```

Out[15]:
```
array([False, False, False, ..., False, False, False])
```

check if any wines value is True for the condition quality > 7

In [16]:
```
check=wine[:,11]>6
True in check
```

Out[16]:
```
True
```

Show first 3 rows where wine quality > 7, call it high_quality

In [17]:
```
high_Quality=wine[:,11]>7
high_Quality
```

Out[17]:
```
array([False, False, False, ..., False, False, False])
```

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

Out[18]:
```
0.56
```

Show only top 3 rows and all columns of high_quality wines data

In [19]:
```
r=wine[high_Quality][0:3]
r
```

Out[19]:
```
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]])
```

Show wines with a lot of alcohol > 10 and high wine quality > 7

In [20]:
```
d = (wine[:,10]>10) & (wine[:,11]>7)
d
```

Out[20]:
```
array([False, False, False, ..., False, False, False])
```

show only alcohol and wine quality columns

In [21]:
```
wine[d,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]:
```
wine2=num.genfromtxt("C:/Users/user/Downloads/winequality-white.csv",delimiter=";",skip_header=1)
wine2
```

Out[22]:
```
array([[ 7.  ,  0.27,  0.36, ...,  0.45,  8.8 ,  6.  ],
       [ 6.3 ,  0.3 ,  0.34, ...,  0.49,  9.5 ,  6.  ],
       [ 8.1 ,  0.28,  0.4 , ...,  0.44, 10.1 ,  6.  ],
       ...,
       [ 6.5 ,  0.24,  0.19, ...,  0.46,  9.4 ,  6.  ],
       [ 5.5 ,  0.29,  0.3 , ...,  0.38, 12.8 ,  7.  ],
       [ 6.  ,  0.21,  0.38, ...,  0.32, 11.8 ,  6.  ]])
```

Show size of white_wines

In [23]:
```
wine2.shape[0]
```

Out[23]:
```
4898
```

combine wines and wine2 using vstack3

In [24]:
```
all_wine=num.vstack((wine,wine2))
```

In [25]:
```
all_wine.shape
```

Out[25]:
```
(6497, 12)
```

combine wine and wine2 using concatenate

In [26]:
```
winecanc=num.concatenate((wine,wine2),axis=0)
winecanc
```

Out[26]:
```
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.  ],
       ...,
       [ 6.5 ,  0.24,  0.19, ...,  0.46,  9.4 ,  6.  ],
       [ 5.5 ,  0.29,  0.3 , ...,  0.38, 12.8 ,  7.  ],
       [ 6.  ,  0.21,  0.38, ...,  0.32, 11.8 ,  6.  ]])
```

Matrix Opearation and Reshape

find transpose wines

In [27]:
```
wt=wine.T
wt.shape
```

Out[27]:
```
(12, 1599)
```

Convert wines data into 1D array

In [28]:
```
wr=wine.ravel()
wr
```

Out[28]:
```
array([ 7.4 ,  0.7 ,  0.  , ...,  0.66, 11.  ,  6.  ])
```

In [29]:
```
wr.shape
```

Out[29]:
```
(19188,)
```

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

In [30]:
```
wine[0].reshape(2,6)
```

Out[30]:
```
array([[ 7.4   ,  0.7   ,  0.    ,  1.9   ,  0.076 , 11.    ],
       [34.    ,  0.9978,  3.51  ,  0.56  ,  9.4   ,  5.    ]])
```

Sort alcohol column Ascending Order

In [31]:
```
sorted_alcohol=num.sort(wine[:,10])
sorted_alcohol
```

Out[31]:
```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```

Make sorting to take place in-place

In [32]:
```
wine[:,10].sort()
wine[:,10]
```

Out[32]:
```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```

Will original data be modified?. Check top 10 rows

In [33]:
```
sorted_alcohol_desc=num.sort(wine[:,10])[::-1]
sorted_alcohol_desc
```

Out[33]:
```
array([14.9, 14. , 14. , ...,  8.5,  8.4,  8.4])
```

Will original data be modified?. Check top 10 rows

In [34]:
```
wine[:,10]
```

Out[34]:
```
array([ 8.4,  8.4,  8.5, ..., 14. , 14. , 14.9])
```