

```
#Name:Vivian Richards W  
#Roll no: 205229133
```

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

```
In [12]: wines = np.genfromtxt("winequality-white.csv", delimiter=";", skip_header=1)
```

## 0.1.1 NumPy Aggregation Methods

### Find sum of all residual sugar values

```
In [13]: wines[:, 3].sum()
```

```
Out[13]: 31305.15
```

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

```
In [14]: wines.sum(axis=0)
```

```
Out[14]: array([3.35747500e+04, 1.36282500e+03, 1.63687000e+03, 3.13051500e+04,  
                2.24193000e+02, 1.72939000e+05, 6.77690500e+05, 4.86874609e+03,  
                1.56161300e+04, 2.39927000e+03, 5.14988800e+04, nan])
```

### Find sum of every row

```
In [15]: wines.sum(axis=1)
```

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

```
In [16]: wines.sum(axis=1).shape
```

```
Out[16]: (4898,)
```

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

```
In [17]: wines[:,3].astype(int)
```

```
Out[17]: array([20,  1,  6, ...,  1,  1,  0])
```

```
In [18]: np.max(wines[:,3].astype(int))
```

```
Out[18]: 65
```

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

```
In [19]: np.min(wines[:,3].astype(int))
```

```
Out[19]: 0
```

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

```
In [20]: np.mean(wines[:,3])
```

```
Out[20]: 6.391414863209474
```

## What is 25 percentile residual sugar value?

```
In [21]: np.percentile(wines[:,3], 25)
```

```
Out[21]: 1.7
```

## What is 75 percentile residual sugar value?

```
In [22]: np.percentile(wines[:,3], 75)
```

```
Out[22]: 9.9
```

## Find the average of each feature value

```
In [23]: wines.mean(axis=0)
```

```
Out[23]: array([6.85478767e+00, 2.78241119e-01, 3.34191507e-01, 6.39141486e+00,
                4.57723561e-02, 3.53080849e+01, 1.38360657e+02, 9.94027376e-01,
                3.18826664e+00, 4.89846876e-01, 1.05142670e+01, nan])
```

## 0.1.2 NumPy Array Comparisons

### Show all wines with quality > 5

```
In [24]: wines[:, 11] > 5
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: RuntimeWarning: invalid value encountered in greater  
    """Entry point for launching an IPython kernel.
```

```
Out[24]: array([False, False, False, ..., False, False, False])
```

```
In [25]: wines[:, 11] > 7
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: RuntimeWarning: invalid value encountered in greater  
    """Entry point for launching an IPython kernel.
```

```
Out[25]: array([False, False, False, ..., False, False, False])
```

```
In [26]: # check if any value is True  
np.any((wines[:, 11] > 7) == True)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: RuntimeWarning: invalid value encountered in greater
```

```
Out[26]: False
```

### Show first 3 rows where wine quality > 7

```
In [28]: high_quality = wines[:, 11] > 7
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: RuntimeWarning: invalid value encountered in greater  
    """Entry point for launching an IPython kernel.
```

```
In [29]: high_quality
```

```
Out[29]: array([False, False, False, ..., False, False, False])
```

```
In [30]: wines[high_quality, :][:3, :]
```

```
Out[30]: array([], shape=(0, 12), dtype=float64)
```

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

```
In [32]: high_quality_and_alcohol = (wines[:,10] > 10) & (wines[:,11] > 7)
# show only alcohol and wine quality columns
wines[high_quality_and_alcohol,10:]
```

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel\_launcher.py:1: RuntimeWarning: invalid value encountered in greater  
 """Entry point for launching an IPython kernel.

```
Out[32]: array([], shape=(0, 2), dtype=float64)
```

## 0.1.3 Combining NumPy

### Arrays Combine red wine and white wine data

#### Open white wine dataset

```
In [35]: white_wines = np.genfromtxt("winequality-white.csv", delimiter=";", skip_header=1)
white_wines.shape
```

```
Out[35]: (4898, 12)
```

```
In [36]: all_wines = np.vstack((wines, white_wines))
all_wines.shape
```

```
Out[36]: (9796, 12)
```

### Combine using concatenate method

```
In [37]: data2 = np.concatenate((wines, white_wines), axis=0)
```

```
In [38]: data2.shape
```

```
Out[38]: (9796, 12)
```

## 0.1.4 Matrix Operations and Reshape

### Transpose wine data

```
In [39]: np.transpose(wines).shape
```

```
Out[39]: (12, 4898)
```

## Convert wine data into 1D array

```
In [40]: wines.ravel()
```

```
Out[40]: array([ 7. ,  0.27,  0.36, ...,  0.32, 11.8 ,   nan])
```

```
In [41]: wines.ravel().shape
```

```
Out[41]: (58776,)
```

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

```
In [42]: wines[1,:].reshape((2,6))
```

```
Out[42]: array([[6.30e+00, 3.00e-01, 3.40e-01, 1.60e+00, 4.90e-02, 1.40e+01],  
               [1.32e+02, 9.94e-01, 3.30e+00, 4.90e-01, 9.50e+00,   nan]])
```

## 0.1.5 Sort alcohol column Ascending Order

```
In [44]: sorted_alcohol = np.sort(wines[:, 10])
```

```
In [46]: sorted_alcohol
```

```
Out[46]: array([ 8. ,  8. ,  8.4 , ..., 14. , 14.05, 14.2 ])
```

```
In [47]: # In-place sorting  
wines[:, 10].sort()
```

```
In [48]: wines[:, 10]
```

```
Out[48]: array([ 8. ,  8. ,  8.4 , ..., 14. , 14.05, 14.2 ])
```

## 0.1.6 Sort alcohol column Descending Order

```
In [49]: sorted_alcohol_desc = np.sort(wines[:, 10])[::-1]
```

```
In [50]: sorted_alcohol_desc
```

```
Out[50]: array([14.2 , 14.05, 14.  , ...,  8.4 ,  8.  ,  8.  ])
```

```
In [51]: # original data not modified  
wines[:, 10]
```

```
Out[51]: array([ 8.  ,  8.  ,  8.4 , ..., 14.  , 14.05, 14.2 ])
```

```
In [ ]:
```