Objective: This worksheet shows how to load a Scikit-Learn (sklearn) data set into the Python environment as Pandas DataFrame and perform few operations on it. These <u>sklearn data sets</u> are popularly called the Toy Data Sets, which are primarily used for academic and research purpose.

• A dataset for wines will be loaded into the Python environment from sklearn. This data set contains *classification data* to classify the wines into 3 classes based on few features (attributes).

```
from sklearn.datasets import load_wine
import numpy as np
import pandas as pd

wineData = load_wine()
```

• The dataset contains few NumPy nd-arrays. Important ones are data and *target*. The snippet is shown below (zoom-in to see the details):

• For the given data set, the *data* is of size 178x13. There are 178 rows with 13 attributes (excluding the class). The names of these attributes are called *feature names*.

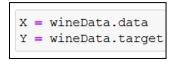
• For the given data set, the *target* is of size 178. There are 178 class values; one for each record. The names of these target are called *target names*. These are the names of different classes.

```
print(type(wineData.target))
print(wineData.target.shape)
print (wineData.target_names)

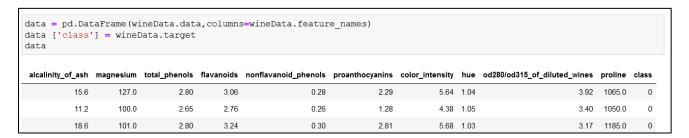
<class 'numpy.ndarray'>
(178,)
['class_0' 'class_1' 'class_2']
```



 Many times in the area of Data Science, all attributes except the class is called the X attribute and the class or the target attribute is called the Y class. They can be assigned as follows before processing further:



• The wine data which is stored in the variable wineData can be transported into Pandas DataFrames as shown below. First a data frame is created using the data part of the data set and then the class is added from the target.

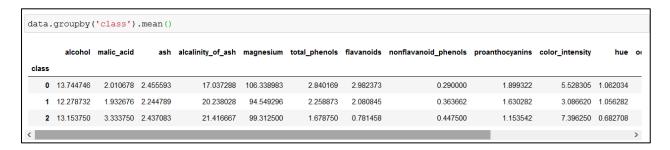


- Portion of the data frame is shown above. You would appreciate that is appears more organized.
- The unique count of different classes can be summarised below using the *value_counts ()* function. This is helpful if one wants to know how many records are present in the classes.

```
print (data['class'].value_counts())

1   71
0   59
2   48
Name: class, dtype: int64
```

 There is a function called groupby () that can be used to group the data and perform some calculations based on that.



 Note that the grouping is done based on the 3 classes and then the mean for each class per attribute is shown.

Exercise:

Load **breast_cancer** classification data from sklearn and convert that into a Pandas data frame. How many classes are there? How many rows in each class? What are the class names?