```
import pandas as pd
import csv
import numpy as np
```

1. Down load the dataset winequality-red.csv file(each column is separated by a semicolon (;)) from the UCI Machine Learning Repository

```
wines = pd.read_csv('C:\\Users\\exam2\\Downloads\\winequality-red.csv')
```

2. Convert it to numPy array, name it as wines (leave the first row of the list) and specify the data type of array as float.

```
wines = np.genfromtxt('C:\\Users\\exam2\\Downloads\\winequality-red.csv',dtype=float,delimiter=';',skip_header=1)
print(wines)
```

Output:

```
[[ 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.   ]]
```

3. Identify the shape of the array.

```
wines.shape
```

Output:

```
(1599, 12)
```

4. Display the element at row 3 and column 4.

```
wines[3][4]
```

Output:

```
0.075
```

5. Display the first three items from the fourth column.

```
print(wines[:3,3])
```

Output:

```
[1.9 2.6 2.3]
```

6. Display third column from each row.

print(wines[:,2])

Output:

```
[0.   0.   0.04 ... 0.13 0.12 0.47]
```

7. Display fourth row.

print(wines[3])

Output:

```
[11.2   0.28  0.56  1.9   0.075 17.  60.  0.998  3.16 0.58  9.8  6.]
```

8. Assign value 10 to 2nd row and 6th column element.

```
wines[1,5] = 10
```

9. Take the 10th column from wines array and name that slice as slice_new and assign value 666 to all elements of slice_new.

```
slice_new = wines[:,9]
slice_new[:] = 666
slice_new
```

Output:array([666., 666., 666., ..., 666., 666., 666.])

10. Display wines array.

```
print(wines)
```

Output:

```
[[7.40e+00 7.00e-01 0.00e+00 ... 6.66e+02 9.40e+00 5.00e+00]
 [7.80e+00 8.80e-01 0.00e+00 ... 6.66e+02 9.80e+00 5.00e+00]
 [7.80e+00 7.60e-01 4.00e-02 ... 6.66e+02 9.80e+00 5.00e+00]
 ...
 [6.30e+00 5.10e-01 1.30e-01 ... 6.66e+02 1.10e+01 6.00e+00]
 [5.90e+00 6.45e-01 1.20e-01 ... 6.66e+02 1.02e+01 5.00e+00]
 [6.00e+00 3.10e-01 4.70e-01 ... 6.66e+02 1.10e+01 6.00e+00]]
```

11 .Find the data type of wines array and Change the data type to int.

```
print(wines.dtype)
wines.astype(int)
```

Output:

```
float64
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]])
```

12. Add 10 points to each quality score.

```
wines[:,-1] += 10

wines
```

```
array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  , 15.   ],
       [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  , 15.   ],
       [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  , 15.   ],
       ...,
       [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   , 16.   ],
       [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  , 15.   ],
       [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   , 16.   ]])
```

13. Find the sum of all the elements in an array

```
print('sum of array : ',np.sum(wines))
```

Output:

sum of array : 152084.78194

14. Find the sum of all the values in every column.

```
print('row wise : ',np.sum(wines,axis = 1))
```

Output:

```
row wise :  [74.5438 123.0548 99.699 ... 100.48174 105.21547  92.49249]
```

15. Find the sum of all the values in every row.

```
print('column wise : ',np.sum(wines,axis=0))
```

Output:

column wise : [13303.1 843.985 433.29 4059.55 139.859 25384.

```
 74302.     1593.79794  5294.47     1052.38    16666.35     9012.  ]
```

16. Add the quality column to itself.

```
wines[:,-1] += wines[:,-1]
wines
```

Output:

```
array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  , 10.   ],
       [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  , 10.   ],
       [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  , 10.   ],
       ...,
       [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   , 12.   ],
       [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  , 10.   ],
       [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   , 12.   ]])
```

17. Multiply alcohol by quality.

```
wines[:,-2] *= wines[:,-1]
print(wines)
```

Output:

```
[[7.40e+00 7.00e-01 0.00e+00 ... 5.60e-01 9.40e+01 1.00e+01]
```

```
   [7.80e+00 8.80e-01 0.00e+00 ... 6.80e-01 9.80e+01 1.00e+01]
 [7.80e+00 7.60e-01 4.00e-02 ... 6.50e-01 9.80e+01 1.00e+01]
 ...
 [6.30e+00 5.10e-01 1.30e-01 ... 7.50e-01 1.32e+02 1.20e+01]
 [5.90e+00 6.45e-01 1.20e-01 ... 7.10e-01 1.02e+02 1.00e+01]
 [6.00e+00 3.10e-01 4.70e-01 ... 6.60e-01 1.32e+02 1.20e+01]]
```

18. Display which wines have a quality rating higher than 5.

```
np.where(wines[:,-1]>5)
```

```
(array([ 3,    7,    8,   16,   19,   20,   24,   29,   31,   33, 35,
         36,   37,   42,   51,   52,   54,   59,   62,   69,   70, 77,
         84,   86,   91,   95,   99,  100,  101,  102,  108,  113, 115,
         116,  117,  118,  119,  121,  128,  133,  134,  142,  144,148,
         149,  150,  159,  162,  168,  171,  172,  173,  177,  184,191,
         197,  198,  200,  204,  205,  206,  209,  210,  211,  212,214,
         220,  223,  225,  226,  228,  230,  231,  232,  234,  235,236,
         237,  238,  239,  241,  242,  243,  244,  245,  248,  249,250,
         251,  254,  259,  265,  267,  268,  269,  270,  271,  275,276,
         277,  278,  279,  280,  281,  283,  286,  287,  288,  290,292,
         293,  294,  300,  301,  305,  307,  308,  309,  310,  311,312,
         315,  317,  318,  319,  320,  323,  324,  325,  326,  328,330,
         331,  332,  334,  335,  336,  338,  339,  340,  341,  342,343,
         344,  346,  347,  348,  349,  350,  351,  354,  355,  357,358,
         359,  361,  364,  365,  366,  369,  371,  372,  374,  375,376,
         377,  378,  379,  380,  381,  382,  383,  385,  386,  387,388,
         389,  390,  391,  395,  397,  398,  401,  402,  403,  405,406,
         407,  408,  410,  413,  416,  418,  420,  421,  423,  425,426,
         427,  429,  430,  432,  434,  436,  437,  438,  440,  441,442,
         443,  444,  445,  448,  449,  450,  451,  452,  453,  455,458,
         460,  464,  466,  467,  468,  471,  472,  474,  477,  479,481,
       ……..
       1042, 1043, 1044, 1045, 1046, 1048, 1049, 1053, 1054, 1055,1056,
       1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067,1068
       1070, 1072, 1073, 1075, 1076, 1079, 1080, 1081, 1082, 1083,1084,
       1086, 1087, 1088, 1089, 1090, 1091, 1092, 1093, 1094, 1096,1098,
       1100, 1101, 1102, 1103, 1104, 1106, 1107, 1109, 1110, 1111,1112,
       1113, 1114, 1115, 1116, 1117, 1118, 1120, 1121, 1122, 1123,1125,
       1126, 1127, 1129, 1130, 1132, 1133, 1134, 1135, 1136, 1137,1139,
       1140, 1141, 1142, 1143, 1145, 1146, 1147, 1148, 1149, 1150,1151
       1153, 1154, 1156, 1157, 1158, 1160, 1161, 1162, 1167, 1168,1169,
       1170, 1171, 1172, 1173, 1174, 1175, 1177, 1179, 1180, 1182,1185,
       1187, 1190, 1192, 1194, 1195, 1196, 1197, 1198, 1199, 1200,1201,
       1202, 1204, 1205, 1206, 1208, 1209, 1210, 1212, 1213, 1214,1215,
       1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1228,1230,
       1234, 1236, 1237, 1242, 1244, 1248, 1249, 1250, 1257, 1258,1259,
       1264, 1265, 1266, 1267, 1268, 1269, 1270, 1271, 1274, 1275,1277,
       1278, 1279, 1280, 1281, 1282, 1283, 1286, 1291, 1292, 1294,1297,
       1298, 1300, 1301, 1302, 1311, 1313, 1314, 1315, 1316, 1317,1318,
       1319, 1321, 1323, 1324, 1325, 1326, 1327, 1329, 1330, 1332,1335,
       1339, 1340, 1341, 1342, 1343, 1345, 1351, 1357, 1359, 1362,1364,
       1367, 1368, 1371, 1377, 1378, 1379, 1380, 1390, 1395, 1398,1399,
       1402, 1403, 1404, 1405, 1406, 1407, 1408, 1409, 1410, 1411,1412,
       1417, 1422, 1424, 1425, 1426, 1431, 1432, 1433, 1434, 1435,1439,
       1440, 1441, 1444, 1445, 1449, 1450, 1451, 1452, 1454, 1455,1456
       1459, 1460, 1462, 1463, 1466, 1468, 1472, 1475, 1477, 1489,1490,
       1494, 1495, 1497, 1498, 1499, 1503, 1504, 1506, 1507, 1508,1510,
       1512, 1513, 1514, 1515, 1517, 1520, 1524, 1526, 1527, 1528,1529,
```

```
       1530, 1532, 1534, 1535, 1536, 1537, 1540, 1541, 1542, 1543,1544,
       1545, 1549, 1552, 1554, 1555, 1557, 1565, 1566, 1569, 1570,1571,
       1573, 1574, 1575, 1576, 1577, 1578, 1580, 1584, 1585, 1586,1587,
1588, 1590, 1591, 1592, 1593, 1595, 1596, 1598], dtype=int64),)
```

19. Check if any wines have a quality rating equal to 10.

```
np.where(wines[:,-1]==10)
(array([], dtype=int64),)
```

20. Select rows in wines where the quality is over 7

```
wines[wines[:,-1]>7]
array([[7.9000e+00, 3.5000e-01, 4.6000e-01, 3.6000e+00, 7.8000e-02,
        1.5000e+01, 3.7000e+01, 9.9730e-01, 3.3500e+00, 8.6000e-01,
        1.2800e+01, 8.0000e+00],
       [1.0300e+01, 3.2000e-01, 4.5000e-01, 6.4000e+00, 7.3000e-02,
        5.0000e+00, 1.3000e+01, 9.9760e-01, 3.2300e+00, 8.2000e-01,
        1.2600e+01, 8.0000e+00],
       [5.6000e+00, 8.5000e-01, 5.0000e-02, 1.4000e+00, 4.5000e-02,
        1.2000e+01, 8.8000e+01, 9.9240e-01, 3.5600e+00, 8.2000e-01,
        1.2900e+01, 8.0000e+00],
       [1.2600e+01, 3.1000e-01, 7.2000e-01, 2.2000e+00, 7.2000e-02,
        6.0000e+00, 2.9000e+01, 9.9870e-01, 2.8800e+00, 8.2000e-01,
        9.8000e+00, 8.0000e+00],
       [1.1300e+01, 6.2000e-01, 6.7000e-01, 5.2000e+00, 8.6000e-02,
        6.0000e+00, 1.9000e+01, 9.9880e-01, 3.2200e+00, 6.9000e-01,
        1.3400e+01, 8.0000e+00],
       [7.8000e+00, 5.7000e-01, 9.0000e-02, 2.3000e+00, 6.5000e-02,
        3.4000e+01, 4.5000e+01, 9.9417e-01, 3.4600e+00, 7.4000e-01,
        1.2700e+01, 8.0000e+00],
       [8.6000e+00, 4.2000e-01, 3.9000e-01, 1.8000e+00, 6.8000e-02,
        6.0000e+00, 1.2000e+01, 9.9516e-01, 3.3500e+00, 6.9000e-01,
        1.1700e+01, 8.0000e+00],
       [5.5000e+00, 4.9000e-01, 3.0000e-02, 1.8000e+00, 4.4000e-02,
        2.8000e+01, 8.7000e+01, 9.9080e-01, 3.5000e+00, 8.2000e-01,
        1.4000e+01, 8.0000e+00],
       [7.2000e+00, 3.3000e-01, 3.3000e-01, 1.7000e+00, 6.1000e-02,
        3.0000e+00, 1.3000e+01, 9.9600e-01, 3.2300e+00, 1.1000e+00,
        1.0000e+01, 8.0000e+00],
       [7.4000e+00, 3.6000e-01, 3.0000e-01, 1.8000e+00, 7.4000e-02,
        1.7000e+01, 2.4000e+01, 9.9419e-01, 3.2400e+00, 7.0000e-01,
        1.1400e+01, 8.0000e+00]])
```

21. Display wines with alcohol greater than 10 and quality greater than 7

```
wines[np.where((wines[:,-2]>10) & (wines[:,-1]>7))]

array([[7.9000e+00, 3.5000e-01, 4.6000e-01, 3.6000e+00, 7.8000e-02,
        1.5000e+01, 3.7000e+01, 9.9730e-01, 3.3500e+00, 8.6000e-01,
        1.2800e+01, 8.0000e+00],
       [1.0300e+01, 3.2000e-01, 4.5000e-01, 6.4000e+00, 7.3000e-02,
```

```
        5.0000e+00, 1.3000e+01, 9.9760e-01, 3.2300e+00, 8.2000e-01,
        1.2600e+01, 8.0000e+00],
       [5.6000e+00, 8.5000e-01, 5.0000e-02, 1.4000e+00, 4.5000e-02,
        1.2000e+01, 8.8000e+01, 9.9240e-01, 3.5600e+00, 8.2000e-01,
        1.2900e+01, 8.0000e+00],
       [1.1300e+01, 6.2000e-01, 6.7000e-01, 5.2000e+00, 8.6000e-02,
        6.0000e+00, 1.9000e+01, 9.9880e-01, 3.2200e+00, 6.9000e-01,
        1.3400e+01, 8.0000e+00],
       [5.0000e+00, 4.2000e-01, 2.4000e-01, 2.0000e+00, 6.0000e-02,
        1.9000e+01, 5.0000e+01, 9.9170e-01, 3.7200e+00, 7.4000e-01,
        1.4000e+01, 8.0000e+00],
       [7.8000e+00, 5.7000e-01, 9.0000e-02, 2.3000e+00, 6.5000e-02,
        3.4000e+01, 4.5000e+01, 9.9417e-01, 3.4600e+00, 7.4000e-01,
        1.2700e+01, 8.0000e+00],
       [9.1000e+00, 4.0000e-01, 5.0000e-01, 1.8000e+00, 7.1000e-02,
        7.0000e+00, 1.6000e+01, 9.9462e-01, 3.2100e+00, 6.9000e-01,
        1.2500e+01, 8.0000e+00],
       [1.0000e+01, 2.6000e-01, 5.4000e-01, 1.9000e+00, 8.3000e-02,
        4.2000e+01, 7.4000e+01, 9.9451e-01, 2.9800e+00, 6.3000e-01,
        1.1800e+01, 8.0000e+00],
       [7.9000e+00, 5.4000e-01, 3.4000e-01, 2.5000e+00, 7.6000e-02,
        8.0000e+00, 1.7000e+01, 9.9235e-01, 3.2000e+00, 7.2000e-01,
        1.3100e+01, 8.0000e+00],
       [8.6000e+00, 4.2000e-01, 3.9000e-01, 1.8000e+00, 6.8000e-02,
        6.0000e+00, 1.2000e+01, 9.9516e-01, 3.3500e+00, 6.9000e-01,
        1.1700e+01, 8.0000e+00],
       [5.5000e+00, 4.9000e-01, 3.0000e-02, 1.8000e+00, 4.4000e-02,
        2.8000e+01, 8.7000e+01, 9.9080e-01, 3.5000e+00, 8.2000e-01,
        1.4000e+01, 8.0000e+00],
       [7.2000e+00, 3.8000e-01, 3.1000e-01, 2.0000e+00, 5.6000e-02,
        1.5000e+01, 2.9000e+01, 9.9472e-01, 3.2300e+00, 7.6000e-01,
        1.1300e+01, 8.0000e+00],
       [7.4000e+00, 3.6000e-01, 3.0000e-01, 1.8000e+00, 7.4000e-02,
        1.7000e+01, 2.4000e+01, 9.9419e-01, 3.2400e+00, 7.0000e-01,
        1.1400e+01, 8.0000e+00]])
```

22. Change the shape of wines array.

wines.reshape(12,1599)

Output:

```
array([[7.40000e+00, 7.00000e-01, 0.00000e+00, ..., 6.60000e+00,
        5.00000e-01, 1.00000e-02],
       [1.50000e+00, 6.00000e-02, 1.70000e+01, ..., 3.30000e+00,
        9.60000e-02, 2.60000e+01],
       [6.10000e+01, 1.00025e+00, 3.60000e+00, ..., 4.20000e+01,
        9.96300e-01, 3.10000e+00],
       ...,
       [2.30000e+00, 7.60000e-02, 2.30000e+01, ..., 1.70000e+00,
        7.50000e-02, 6.00000e+00],
       [2.50000e+01, 9.95810e-01, 3.09000e+00, ..., 5.30000e+01,
        9.95800e-01, 3.41000e+00],
       [6.70000e-01, 9.70000e+00, 5.00000e+00, ..., 6.60000e-01,
        1.10000e+01, 6.00000e+00]])
```